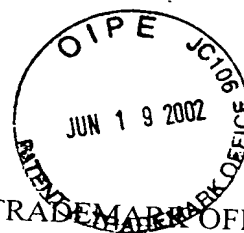


Docket No.: 163-386

COPY OF PAPERS
ORIGINALLY FILED



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
PATENT OPERATIONS

In re Application of:

Giovanni Bazzoni, et al

Serial No.: 10/099,856

Filed: March 14, 2002

)
)
) Group Art Unit: --
)
) Examiner: --
)
)
)

For: **SYSTEM FOR THE CREATION, VISUALISATION AND MANAGEMENT OF
THREE-DIMENSIONAL OBJECTS ON WEB PAGES AND A RELATIVE METHOD**

New York, NY 10036
June 14, 2002

Commissioner for Patents
Washington, DC 20231

CLAIM FOR CONVENTION PRIORITY UNDER 35 U.S.C. §119

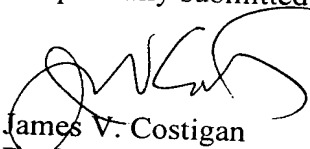
SIR:

In the matter of the above-identified application and under the provisions of 35
U.S.C. §119 Inventor(s) claim the benefit of the following prior applications:

Application(s) filed in	:	Italy
In the name of Applicant(s)	:	Giovanni Bazzoni, et al
Application No(s).	:	MI 2001 A 000538
Filed	:	March 14, 2001

Pursuant to the Claim to Priority, Applicant(s) submit duly certified copy of
said foreign application.

Respectfully submitted,

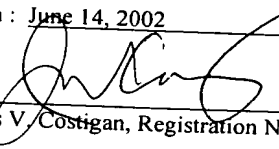

James V. Costigan
Registration No. 25,669

HEDMAN & COSTIGAN, P.C.
1185 Avenue of the Americas
New York, NY 10036-2646
(212) 302-8989

I hereby certify that this correspondence is being
deposited with the United States Postal Service
as first class mail in an envelope addressed to:

Commissioner for Patents
Washington, DC 20231

on : June 14, 2002

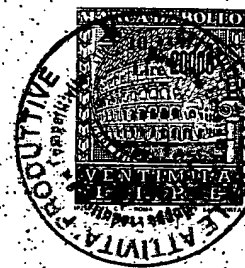

James V. Costigan, Registration No. 25,669



Serial # 10/099,856

Mod. C.E. - 1-4-7

Ministero delle Attività Produttive
Direzione Generale per lo Sviluppo Produttivo e la Competitività
Ufficio Italiano Brevetti e Marchi
Ufficio G2



Autenticazione di copia di documenti relativi alla domanda di brevetto per:

Invenzione Industriale

N.

MI2001 A 000538

*Si dichiara che l'unità copia è conforme ai documenti originali
depositati con la domanda di brevetto soprastipificata, i cui dati
risultano dall'accluso processo verbale di deposito.*

BEST AVAILABLE COPY

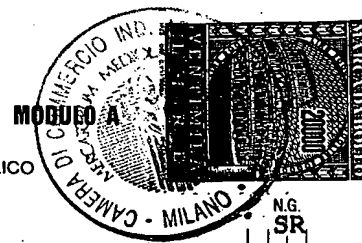
Ron

10 APR. 2002

IL DIRIGENTE

Elena Marinelli
Sig.ra E. MARINELLI

AL MINISTERO DELL'INDUSTRIA DEL COMMERCIO E DELL'ARTIGIANATO
UFFICIO ITALIANO BREVETTI E MARCHI - ROMA
DOMANDA DI BREVETTO PER INVENZIONE INDUSTRIALE, DEPOSITO RISERVE, ANTICIPATA ACCESSIBILITÀ AL PUBBLICO



A. RICHIEDENTE (I)

1) Denominazione **PHOENIX TOOLS S.R.L.**
 Residenza **MILANO** codice **12169840159**
 2) Denominazione _____
 Residenza _____ codice _____

B. RAPPRESENTANTE DEL RICHIEDENTE PRESSO L'U.I.B.M.

cognome nome **BURCHIELLI Riccardo e altri** cod. fiscale _____
 denominazione studio di appartenenza **BARZANO' & ZANARDO MILANO S.p.A.**
 via **BORGONUOVO** n. **10** città **MILANO** cap **20121** (prov) **MI**

C. DOMICILIO ELETTIVO destinatario

via _____ n. _____ città _____ cap _____ (prov) _____

D. TITOLO

classe proposta (sez/cl/sci) _____ gruppo/sottogruppo _____

SISTEMA PER LA CREAZIONE, LA VISUALIZZAZIONE E LA GESTIONE DI OGGETTI TRIDIMENSIONALI SU PAGINE WEB E METODO RELATIVO

ANTICIPATA ACCESSIBILITÀ AL PUBBLICO:

SI ☐ NO ☒

SE ISTANZA: DATA _____

N° PROTOCOLLO _____

E. INVENTORI DESIGNATI

cognome nome

cognome nome

1) **BAZZONI GIOVANNI** 3) **RINALDI GERARDO**
 2) **RICCARDI KARIM** 4) **SAVARE' RICCARDO**

F. PRIORITÀ

nazione o organizzazione	tipo di priorità	numero di domanda	data di deposito	allegato S/R
1) _____	_____	_____	____/____/____	_____
2) _____	_____	_____	____/____/____	_____

G. CENTRO ABILITATO DI RACCOLTA CULTURE DI MICROORGANISMI, denominazione

H. ANNOTAZIONI SPECIALI

DOCUMENTAZIONE ALLEGATA

N. es.

Doc. 1) L2 PROV n. pag. L7D	riassunto con disegno principale, descrizione e rivendicazioni (obbligatorio 1 esemplare)
Doc. 2) L2 PROV n. tav. L12	disegno (obbligatorio se citato in descrizione, 1 esemplare)
Doc. 3) L0 RIS	lettera d'incarico, procura o riferimento procura generale
Doc. 4) L0 RIS	designazione inventore
Doc. 5) L RIS	documenti di priorità con traduzione in italiano
Doc. 6) L RIS	autorizzazione o atto di cessione
Doc. 7) L	nominativo completo del richiedente

8) attestati di versamento, totale lire **NOVECENTOQUINDICIMILA** obbligatorio
 COMPILATO IL **1/4/03** FIRMA DEL(I) RICHIEDENTE(I) **I MANDATARI (firma per sé e per gli altri)**
 CONTINUA SI/NO **NO**
 DEL PRESENTE ATTO SI RICHIEDE COPIA AUTENTICA SI/NO **SI**

CAMERA DI COMMERCIO IND. ART. E AGR. DI MILANO codice **15**
VERBALE DI DEPOSITO NUMERO DI DOMANDA **MI2001A 000538** Reg. A.
 L'anno **DUEMILAUNO** **QUATTORDICI** del mese di **MARZO**

Il(i) richiedente(i) sopraindicato(i) ha(hanno) presentato a me sottoscritto la presente domanda, corredata di n. _____ fogli aggiuntivi per la concessione del brevetto soprariportato.

I. ANNOTAZIONI VARIE DELL'UFFICIALE ROGANTE

IL DEPOSITANTE

[Signature]



UFFICIALE ROGANTE

M. CORTONESI

RIASSUNTO INVENZIONE CON DISEGNO PRINCIPALE DESCRIZIONE E RIVENDICAZIONE

NUMERO DOMANDA

M/2001A000538

REG. A

DATA DI DEPOSITO

14/03/2001

NUMERO BREVETTO

DATA DI RILASCIO

/ /

D. TITOLO

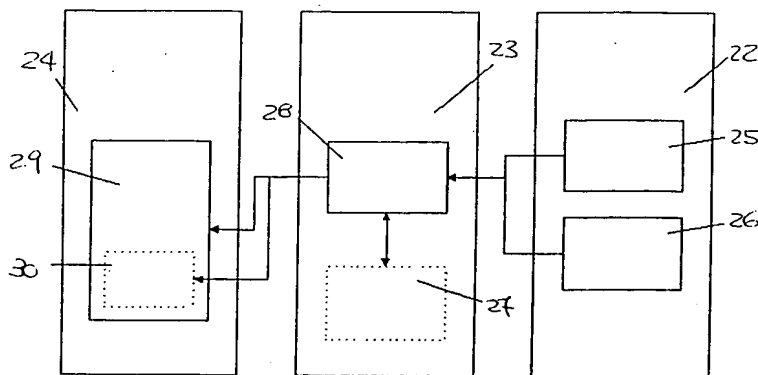
"Sistema per la creazione, la visualizzazione e la gestione di oggetti tridimensionali su pagine web e metodo relativo".

L. RIASSUNTO

Un sistema per la creazione, la visualizzazione e la gestione di oggetti tridimensionali su pagine web, atto a consentire di accedere ad oggetti tridimensionali presenti su un browser web del World-Wide-Web (WWW), che vengono trattati in modo tale da apparire oggetti reali animati sulle pagine standard del web, mentre l'utilizzatore può liberamente muovere, ruotare e generalmente interagire con i suddetti oggetti per visualizzare meglio le informazioni presenti sulle superfici esterne. La principale caratteristica della presente invenzione risiede nel fatto di poter sovrapporre pagine html sull'intera superficie dell'oggetto, in modo che ciascuna faccia di tale oggetto si comporti come un browser web di tipo standard, supportandone le funzionalità tipiche.



M. DISEGNO

Fig.5

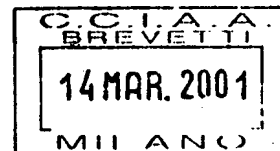
MI 2001A000538

DESCRIZIONE dell'invenzione industriale

a nome: PHOENIX TOOLS S.r.l.

di nazionalità: italiana

con sede in: MILANO (MI).



La presente invenzione si riferisce genericamente alla possibilità di effettuare interazioni, da parte di utenti, nel World-Wide-Web, e, più specificatamente, descrive una apparecchiatura elettronica ed un metodo relativo, che consentono, da una parte, al creatore di siti web, di utilizzare oggetti tridimensionali quali elementi di visualizzazione di informazioni e, d'altra parte, all'utente di tali siti, di interagire con ulteriori contenuti ed applicazioni, visualizzabili in un browser web, per mezzo di interfacce tridimensionali.

In particolare, l'invenzione descrive un metodo per la creazione di oggetti tridimensionali con pagine html visualizzate sulle loro superfici esterne, per la determinazione del grado di interazione con l'utilizzatore e per la definizione delle risposte ai comandi in ingresso dell'utente; l'invenzione descrive inoltre un sistema di visualizzazione di oggetti tridimensionali all'interno di un web browser standard ed una modalità di interazione con l'utente.

Inizialmente, il World-Wide-Web non aveva suscitato molto interesse nella comunità di NSFNet (il nuovo nome dato alla nascente rete dopo l'assorbimento della prima incarnazione di Internet, ARPANet, da parte della National Science Foundation); un motivo era che i siti web erano pochi e bisognava creare degli hyperlink tra i vari siti in modo che questi potessero essere connessi istantaneamente.

La creazione degli hyperlink si otteneva inserendo all'interno del testo di una pagina i mezzi per accedere alle pagine remote, usando il linguaggio HTML ("Hypertext Markup Language"), che costituisce il codice sorgente per il Web; tutto quello che l'utente doveva fare era cliccare sui link per collegarsi attraverso la rete al sito indicato.

Il linguaggio html ha permesso agli utenti di effettuare delle connessioni tra i siti di Internet senza sapere come trovarli; inoltre, gli utenti non dovevano conoscere il linguaggio html, perché i browser web (programmi software che leggono il codice sorgente html e lo convertono in un testo leggibile) si occupavano di tutto questo.

Tuttavia, all'inizio, i siti web si basavano solo sul testo, come i primi browser, e, quindi, non erano né grafici, né eccitanti.

Nel 1993 venne inventato il primo browser grafico gratuito e facile da usare (Mosaic) e ciò accelerò enormemente il traffico sul web e Mosaic divenne l'applicazione per la rete più diffusa.

Con Mosaic e una rete più veloce la grafica diventava una realtà per chiunque avesse accesso a Internet.

Parallelamente, la grafica su computer faceva passi da gigante con l'introduzione di formati di file di immagine standard che ottimizzavano la compressione dei dati per snellire il carico su una banda assai limitata (JPEG).

Nel 1994 la nascita del linguaggio VRML introduceva definitivamente il concetto di strutture tridimensionali in Internet, ipotizzando la possibilità di esplorare mondi virtuali in rete.

Un altro grande evento, nel 1995, fu la conversione della rete da parte di NSFNet in strutture commerciali; da questo momento, le società commerciali di tutto il mondo cominciarono a popolare la rete con i propri siti web per proporre prodotti e servizi, e immisero nel circuito Internet enormi quantità di denaro.

Ciò diede un grande impulso a tutte le società di software che si occuparono a sviluppare applicazioni per il web, non ultime quelle specializzate in grafi-

ca tridimensionale.

Dopo il parziale fallimento del VRML, invece di creare mondi tridimensionali separati dal resto di Internet, l'orientamento fu quello di generare oggetti e ambienti che potessero essere visti e manipolati all'interno di un browser standard.

Questa seconda revisione del concetto di VRML portò le società di software a proporre soluzioni orientate al commercio elettronico e, cioè, a presentare, all'interno di browser web, manufatti visualizzati mediante foto realistiche od immagini di personaggi animati in uno spazio tridimensionale; a tali soluzioni ci si riferisce con il nome di «Player 3D».

I «Player 3D» del web funzionano come i motori di funzionamento dei videogiochi; visualizzano a video delle scene o degli oggetti tridimensionali, li rendono in tempo reale e ne gestiscono l'interattività con l'utente.

Solitamente, gli elementi sono sul server di una società di web tridimensionale e vengono visualizzati tramite dei "plug-in" per il browser che devono essere scaricati ed installati.

Uno dei problemi più grossi in questo campo è l'ampiezza di banda di Internet che impone che gli elementi tridimensionali siano relativamente sempli-



ci; a tale proposito, tutti i produttori «Player 3D» hanno escogitato soluzioni più o meno efficienti per affrontare il problema, in modo da trasferire meno dati possibili al «Player 3D», comunque le tecniche generalmente usate sono la compressione e il trasferimento incrementale dei soli dati cui l'utente sta accedendo.

Anche dal punto di vista del processo di generazione e pubblicazione di un sito web la tecnologia ha fatto passi da gigante.

La diffusione delle società di multimedia che si occupano di fornire soluzioni chiavi in mano per il web alle società, è andata avanti di pari passo con la proliferazione di strumenti evoluti per la generazione delle pagine html; tali strumenti, oltre a facilitare l'inserimento di informazioni testuali e grafiche in una pagina web, consentono l'introduzione e la gestione di funzionalità di interazione con l'utente e, inoltre, tramite i software di "authoring" è possibile produrre interi siti web senza scrivere alcuna riga di codice html, perché è l'applicazione stessa che al momento della pubblicazione provvede a generare il codice html necessario a riprodurre effetti ed informazioni su un normale browser web, in modo tale da consentire all'autore di concentrarsi sull'aspetto

creativo, architettuale e funzionale di una applicazione web.

Sebbene sia ora possibile produrre una pagina Internet ricca di contributi di testo, grafica ed animazioni, il supporto tipicamente bidimensionale offerto da un web browser tradizionale resta comunque un limite invalicabile; infatti, se le informazioni da inserire in un sito web sono tante, l'autore è costretto a distribuire il contenuto su più pagine obbligando l'utente web a seguire i collegamenti (links) che lo portano da una pagina all'altra.

L'alternativa di forzare tutti i contenuti in un'unica grande pagina è generalmente poco praticata per ovvi motivi di estetica e spesso per esigenze di distribuzione delle informazioni in sezioni logiche.

Scopo della presente invenzione è quindi quello di ovviare agli inconvenienti lamentati in precedenza e, in particolare, quello di realizzare un sistema per la creazione, la visualizzazione e la gestione di oggetti tridimensionali su pagine web ed un metodo relativo, che consentano di espandere indefinitamente lo spazio disponibile in una pagina web per immagazzinare informazioni favorendo le caratteristiche estetiche e tecnico-architettoniche del sito web.

Tale scopo è raggiunto da un metodo per la creazione,

la visualizzazione e la gestione di oggetti tridimensionali su pagine web, secondo la rivendicazione 1, e da un sistema implementante tale metodo, secondo la rivendicazione 2, alle quali si rimanda per brevità.

In modo vantaggioso, l'intero sistema oggetto dell'invenzione prevede due attori e due diverse configurazioni "hardware": l'autore delle pagine web e l'utilizzatore del browser web.

Un primo sistema comprende un web server in cui vengono immagazzinate le pagine web contenenti gli oggetti tridimensionali e le chiavi di accesso, un computer per generare le pagine web ed un programma software, mentre un secondo sistema prevede l'esecuzione di web browser su uno o più elaboratori, una applicazione software (il componente attivo «Player 3D») funzionante nel web browser ed il server web, atto all'esecuzione dell'applicazione web.

Quando l'utente accede al server web per scaricare i contenuti del web, utilizza un browser web tradizionale. Il componente attivo «Player 3D», se già presente nel browser web all'interno dell'elaboratore dell'utente, permette l'interpretazione e la visualizzazione di un file di riferimento contenente la descrizione degli oggetti tridimensionali.

Alternativamente, tale componente può essere scarica-

to automaticamente dal server ed installato nell'elaboratore dell'utente. Grazie al componente attivo, l'interazione fra l'utente e gli oggetti tridimensionali generati viene eseguita localmente sul computer dell'utilizzatoree cio' garantisce alte prestazioni e una minimizzazione del carico di rete.

Questo tipo di architettura permette così a numerosi web browser di accedere all'applicazione web simultaneamente.

In pratica, secondo la presente invenzione, si prevede di proporre un sistema ed un metodo per migliorare le soluzioni visive di una applicazione di un web server e per consentire l'accesso ad essa da un web browser del World-Wide-Web, per mezzo di una efficace interfaccia utente.

La presenza di interfacce tridimensionali all'interno di un normale browser web, così come descritte nella presente invenzione, può essere gestita in modalità statica o dinamica.

La dizione statica o dinamica non ha a che fare con le proprietà animate del componente tridimensionale né con le facoltà di interazione da parte dell'utente; essa si riferisce piuttosto alla capacità del componente di auto-generarsi e conformarsi alle esigenze e alle preferenze dell'utente.



Negli ultimi due anni abbiamo assistito ad una evoluzione nell'architettura e nelle funzionalità dei siti web; i siti più complessi spesso si appoggiano a basi di dati ("database") interfacciati con il server web, che, per esempio, conservano gli identificativi degli utenti che possono accedere ad esso e ne personalizzano la navigazione.

Un altro tipico caso di sito dinamico sono i diffusi motori di ricerca che forniscono un servizio interattivo per l'individuazione di indirizzi Internet che corrispondono a parole chiave di ricerca inserite dall'utente.

In tutti questi casi la pagina html visualizzata nel web browser non è staticamente generata dall'autore del sito ma dinamicamente composta dal sistema server a seconda dei casi; tutti i server web sono in grado di eseguire particolari applicazioni (dette "script") (asp, jsp, php, ecc.); esse sono specificatamente sviluppate dal programmatore web per accedere al database, eseguire le query necessarie e generare sulla base delle risposte le adeguate pagine html.

Lo stesso paradigma si applica per la generazione e l'uso dei componenti tridimensionali descritti nella presente trattazione.

L'oggetto tridimensionale visualizzato nella pagina

web può essere staticamente creato e animato dall'autore del sito ed apparire allo stesso modo a tutti gli utenti che accedono a quella determinata pagina; in alternativa, l'oggetto tridimensionale e le pagine web da mappare sulla sua superficie sono generate in maniera dinamica dal programma server eseguito ogniqualvolta si acceda ad una pagina web con tale facoltà.

Per il primo caso è stata progettata una specifica applicazione che facilita l'autore a definire le caratteristiche geometriche e di animazione dell'oggetto che vuole rappresentare in tre dimensioni in una determinata pagina del suo sito e tale applicazione viene definita come sistema autore.

Per il secondo caso ci si affida alla capacità del programmatore del sito affinché implementi in un "server script", i servizi che ritenga necessari.

È compito del server che interpreta lo "script" generare in uscita un file di testo che descrive l'oggetto tridimensionale in un formato interpretabile dal «Player 3D» eseguito sulla macchina dell'utente.

In entrambi i casi l'autore del sito deve preoccuparsi di inserire nel codice html della pagina che vuole ospitare un componente tridimensionale, istruzioni

specifiche per l'invocazione del «Player 3D» e per ottenere le informazioni necessarie sulle caratteristiche dello specifico componente.

In caso di modalità statica, tali informazioni sono costituite, da un file binario generato in maniera automatica dal sistema autore; in caso di modalità dinamica, invece, di un file descrittivo del componente è necessario dichiarare l'applicazione server da eseguire, in modo che sia quest'ultima a restituire il file di descrizione in formato testo.

L'autore del sito web ha, per quanto detto, due modalità per generare e descrivere le proprietà dei componenti tridimensionali che vuole inserire nelle sue pagine: nella prima egli si affida al sistema autore, vale a dire una applicazione software eseguibile da un comune computer, che fornisce una interfaccia grafica "user-friendly" per creare l'oggetto in maniera interattiva e visualizzarne il contenuto in anteprima. All'interno dell'applicazione, l'autore definisce la geometria dell'oggetto e gli indirizzi delle pagine html che vuole visualizzare su ciascuna faccia dell'oggetto; per ogni poligono che costituisce la superficie dell'oggetto, in alternativa alla pagina html, può definire un colore di sfondo.

Una volta definita la geometria e la mappatura

dell'oggetto, il passo successivo consiste nel descriverne l'animazione e i gradi di libertà; l'autore decide, cioè, se, una volta visualizzato sulla pagina web, l'oggetto dovrà traslare e ruotare al suo interno secondo uno schema definito dall'autore e se l'utente potrà intervenire trasportando, ruotando o allontanando e avvicinando l'oggetto col "mouse" a suo piacimento.

Per l'animazione dell'oggetto il sistema fa uso di curve di funzione, secondo cui ad ogni proprietà animabile dell'oggetto è associata una curva, che esprime il valore della proprietà in ogni istante di tempo; un programma "editor" di curve di funzione permette all'autore di definire in modalità grafica la forma di ciascuna curva di funzione nel tempo.

Altre proprietà dell'oggetto selezionabili dall'autore sono descritte in dettaglio in seguito.

A conclusione del processo di definizione dell'oggetto, il sistema autore genera l'uscita necessaria per pubblicare il lavoro fatto sul server web prescelto. L'uscita consiste in un file binario contenente la descrizione dell'oggetto, e da poche righe di codice html da inserire nella codice html della pagina web che deve accogliere l'elemento tridimensionale. Compito di queste ultime è invocare nel



computer dell'utente il componente attivo costituito dal «Player 3D» e passargli il file di descrizione dell'oggetto da visualizzare.

Come detto sopra, in alternativa al sistema autore, il programmatore del sito può implementare applicazioni server la cui esecuzione ha come risultato un file di descrizione dell'oggetto tridimensionale.

A tale scopo, è stato definito un formato proprietario di file, che definisce le regole necessarie affinché la descrizione possa essere interpretata dal «Player 3D»; quest'ultimo è infatti in grado di interpretare indifferentemente sia file binari che testo, purché rispettino il formato descritto.

Alla fine della catena si trova il «Player 3D»: esso è eseguito sul computer dell'utente Internet quando egli accede ad una pagina web che contiene al suo interno una chiamata al «Player 3D».

Se il «Player 3D» non è presente nel computer dell'utente o non è aggiornato, l'utente è invitato a scaricarlo dalla rete; scaricamento ed installazione del «Player 3D» avviene in maniera automatica e trasparente all'utente stesso.

Al momento dell'installazione, il «Player 3D» riceve un descrittore di file (binario o ASCII) contenente la definizione degli oggetti tridimensionali da vi-

sualizzare; le operazioni fondamentali eseguite dal «Player 3D» sono la creazione in memoria di un web browser per ogni faccia dell'oggetto cui è associato l'indirizzo di una pagina web, il caricamento di tali pagine nei web browser in memoria e la trasformazione della pagina in un file immagine, nonché una procedura di "rendering" dell'oggetto con le pagine web mappate sulle sue facce.

Il «Player 3D» si occupa anche di gestire l'interazione con l'utente che, nell'ambito delle eventuali restrizioni di movimento imposte dall'autore, è in grado di manipolare l'oggetto nello spazio del componente attivo che lo contiene.

Il componente attivo è una applicazione Windows che viene eseguita dal web browser ospitante.

La finestra di visualizzazione del componente attivo, le cui dimensioni sono definite dall'autore, si sovrappone alla pagina web sottostante; naturalmente è compito del processo di "rendering" rendere completamente trasparente tutta l'area occupata dal componente attivo, dove non è presente l'oggetto da rappresentare, in modo da non oscurare la pagina sottostante.

Il «Player 3D», oltre a consentire all'utente la facoltà di manipolare la posizione e l'orientamento

dell'oggetto nello spazio, garantisce tutte le comuni operazioni che l'utente è abituato a compiere su una interfaccia basata su "Hypertext".

Come detto, su ciascuna faccia dell'oggetto tridimensionale è possibile mappare intere pagine html, conservandone tutte le peculiarità di interazione con l'utente; tali pagine web rappresentate in maniera prospettica sulla superficie dell'oggetto sono a tutti gli effetti cloni del web browser ospitante e ne replicano le funzionalità.

Ulteriori scopi e vantaggi della presente invenzione risulteranno maggiormente evidenti dalla descrizione che segue, che si riferisce ad un esempio di applicazione preferito ma non limitativo, dell'invenzione e dai disegni allegati, in cui:

- la figura 1 rappresenta un diagramma di flusso della fase di scaricamento del «Player 3D» e del meccanismo di autentica;
- le figure 2a e 2b descrivono i file in formato testo atti a definire un componente tridimensionale riconoscibile da una unità di elaborazione elettronica secondo la presente invenzione;
- la figura 3 mostra alcune regole in formato testo utilizzabili per gli obiettivi definiti nelle figure precedenti;

- la figura 4 mostra un esempio di come descrivere un semplice componente tridimensionale in formato testo, secondo la presente invenzione;
- la figura 5 è uno schema a blocchi delle principali relazioni e dei principali componenti di un procedimento per la creazione e la visualizzazione di oggetti tridimensionali su una pagina web, secondo la presente invenzione;
- la figura 6 mostra un esempio di codice html generato dal sistema di "authoring";
- le figure 7a e 7b mostrano due diagrammi di flusso relativi ad un utilizzo statico del sistema, secondo la presente invenzione;
- le figure 8a e 8b mostrano due diagrammi di flusso relativi ad un utilizzo dinamico del sistema, secondo la presente invenzione;
- la figura 9 è un diagramma di flusso dettagliato del «Player 3D», secondo l'invenzione;
- la figura 10 è un diagramma di flusso dettagliato del processo di "rendering", secondo l'invenzione;
- la figura 11 è un diagramma di flusso dettagliato del sistema di "authoring", secondo l'invenzione.

Con particolare riferimento alla figura 5, che rappresenta lo schema complessivo del sistema hardware e software costituente l'oggetto della presente inven-



zione, si distinguono un lato server 22 ed un lato utente 24 dell'applicazione.

Dal lato server 22 l'autore del sito web è dotato di un computer 22A, atto ad eseguire una applicazione software 25, che costituisce il sistema autore del processo.

Esso è un software dotato di interfaccia grafica "user-friendly", che facilita l'autore nel processo di creazione e di definizione delle proprietà e dei componenti tridimensionali che vuole visualizzare in una pagina web del sito. Nella figura 11 si approfondiranno i meccanismi e gli aspetti peculiari del sistema autore. Il sistema autore genera due risultati fondamentali, costituiti da un file descrittore degli oggetti creati e da una serie di istruzioni html, mostrate in figura 6, da inserire nella pagina html ospitante il componente tridimensionale.

In alternativa al software autore 25, il sistema, secondo l'invenzione, definisce un formato di file di testo con il quale descrivere i componenti tridimensionali e le loro proprietà: trattandosi di dati in chiaro, tale file è compilabile manualmente dall'autore senza affidarsi ad alcun software di supporto (sistema autore 31).

Naturalmente tale operazione non è eseguita a mano ma

programmaticamente e in maniera dinamica, secondo cui il programmatore del sito implementa specifici "server script", per mezzo di un qualsiasi editor di testo 26, che quando eseguiti producono in uscita un file di testo nel formato richiesto (si vedano, a tale proposito, le figure 2a e 2b).

Come ampiamente descritto più avanti a proposito della figura 8, tale modalità, cui ci si riferisce con la definizione di approccio dinamico, in contrapposizione con l'approccio statico basato sull'utilizzo del sistema autore, è particolarmente potente e flessibile: essa offre al programmatore la possibilità di utilizzare la tecnologia proposta come naturale interfaccia di uscita per applicazioni e servizi web che per esempio si appoggiano su una base di dati o database 27.

La pagina html contenente il collegamento al componente tridimensionale, il file che lo descrive e/o la definizione dei server script da eseguire per la generazione in linea dell'oggetto tridimensionale, sono pubblicati sul server web 23, in attesa di essere scaricati da un qualsiasi computer utente 24, che risulta collegato alla rete tramite il browser web 29. Quando si verifica un accesso al server web, il «Player 3D», se non già residente sul computer utente

24, viene scaricato ed eseguito, e il file contenente l'oggetto tridimensionale da visualizzare gli viene presentato in ingresso (fasi 42-45 di figura 7B).

Il componente «Player 3D» 30, come vedremo in dettaglio qui di seguito a proposito della figura 9, viene eseguito localmente sul computer utente 24 e a sua volta creerà in memoria altri cloni del web browser 29 ospitante, da proiettare sulle facce dell'oggetto tridimensionale.

La figura 9 illustra i processi eseguiti dal componente 30 sul computer utente 24; in particolare, essa descrive il metodo di visualizzazione degli elementi tridimensionali su un web browser, secondo la presente invenzione.

Come primo passo, nella fase 1 viene riconosciuto il formato dei dati in ingresso; nel caso in cui si abbia a che fare con un formato testo, il blocco 8 esegue una fase di validazione, utilizzando un file di licenza e verificando essenzialmente che la chiave di accesso proveniente dal server 22 nel documento 2 sia quella relativa al nome del "domain server" 7.

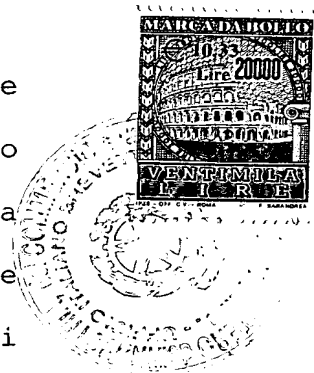
Dopo la fase di validazione, il blocco 3 esegue una lettura della descrizione degli elementi dai documenti 4 o 5 e crea l'oggetto tridimensionale in memoria, pronto ad essere inviato al blocco di processo di

"rendering" 6.

La descrizione dell'elemento contiene informazioni relative alla geometria e all'animazione del componente e gli indirizzi URL di ogni elemento html presente sulle superfici dell'oggetto; i dettagli dei dati trasmessi ai componenti 30 saranno maggiormente chiari da quanto mostrato in figura 2.

I dati principali che vengono letti durante la fase indicata con 3 in figura 9 sono costituiti dal numero di lati della base dell'oggetto tridimensionale, la lunghezza di ciascun lato, le dimensioni delle facce della superficie laterale dell'oggetto, le animazioni ed eventuali stati di riposo preferenziali, i gradi di libertà concessi all'utente.

Il blocco 9 è in grado di realizzare un nuovo web browser 29 per ogni faccia dell'oggetto tridimensionale associato ad un valido URL (vale a dire, l'indirizzo di un file html o di una qualsiasi applicazione server che restituisce un file visualizzabile sul browser). Alcune delle facce che compongono l'oggetto tridimensionale possono anche non avere nessuna associazione di indirizzo URL ma solamente un colore di sfondo. I web browser 29 sono creati nella memoria del computer, vale a dire non sono direttamente visualizzati sull'oggetto tridimensionale, ed il tipo



di browser corrisponde alla stessa applicazione browser che ospita il componente tridimensionale; per esempio, nel caso di un prototipo implementato in ambiente "Internet Explorer", il web browser 29 prescelto può essere un "IWebBrowser" basato su tecnologia COM della Microsoft.

In questa fase inoltre, il blocco 9 registra le relazioni esistenti tra ciascuna faccia dell'oggetto ed il suo relativo web browser 29.

Dopo l'istanziamento dei browser, il blocco 10 carica in ogni web browser 29 l'indirizzo URL associato ad ogni faccia dell'oggetto tridimensionale; mentre viene caricata la pagina, ad istanti predefiniti, il blocco 10 invoca le successive procedure di "texturing" 11 e di "rendering" 6, in modo tale da effettuare una corretta visione dell'oggetto tridimensionale sul monitor 12, in maniera incrementale man mano che la pagina viene scaricata dalla rete.

Quando tutti i browser locali determinano che il caricamento della pagina html è terminato, la procedura di "rendering" 6 viene arrestata e viene iniziata una fase di "rests listening" per il successivo evento 13 scatenato dall'utilizzatore.

Nella fase di "texturing" 11, viene memorizzata l'immagine della pagina html visualizzata nel web browser

29; tale fase 11 rappresenta una delle caratteristiche salienti della presente invenzione.

Una immagine fisica della pagina web e' catturata dal web browser 29 in un determinato istante di tempo, come se si effettuasse una fotografia dello stato corrente dell'informazione visualizzata nel browser. Tale immagine è inviata al blocco di "rendering" 6 insieme con la descrizione geometrica dell'oggetto tridimensionale.

Durante la fase di cattura dell'immagine, tale procedura permette di comunicare con i web browsers funzionanti presenti in memoria, in modo tale da ottenere le posizioni dei "link" attivi all'interno della pagina html.

Le coordinate UV, nel sistema di riferimento dell'immagine html, e le posizioni delle aree rettangolari contenenti i link attivi vengono immagazzinate nel procedimento per elaborare le interazioni dell'utente 13 una volta che la pagina html è stata tradotta in immagine su ciascuna faccia dell'oggetto tridimensionale.

Le immagini da visualizzare su ciascuna faccia, realizzate secondo la procedura precedentemente descritta, insieme con la descrizione geometrica dell'oggetto, vengono inviate nel blocco di "rendering" 6, che

è in grado di visualizzare la struttura dell'oggetto sul monitor 12.

Il processo di "rendering" è basato su un algoritmo a scansione lineare eseguito in tempo reale per massimizzare la velocità di visualizzazione sul monitor 12.

La fase di "rendering" 6 dell'oggetto tridimensionale permette di creare un'immagine dell'oggetto bloccata nel suo movimento in un particolare istante di tempo; l'immagine utilizzata come mappa strutturale di ciascuna faccia dell'oggetto corrisponde alla "bitmap" visualizzata da ogni web browser locale nello stesso istante di tempo e, quindi, l'oggetto visualizzato mostra sulle proprie facce esattamente la medesima immagine che un utente vedrebbe con un browser standard che permette di accedere a quell'indirizzo web.

La fase di "rendering" riceve in ingresso informazioni sulla geometria dell'oggetto tridimensionale da visualizzare, e informazioni sulle trasformazioni spaziali dell'oggetto e le immagini da proiettare sull'oggetto stesso; in particolare, riceve in ingresso le coordinate dei vertici di ciascun poligono insieme con l'immagine da proiettare su ciascuno di essi .

Per individuare i "pixel" in cui ogni linea

dell'immagine in uscita (il "buffer frame rgb") interseca i lati dei poligoni dell'oggetto e da qui calcolare l'intensità di colore di ciascun "pixel" dell'immagine, il processo di "rendering" 6 si basa essenzialmente su un algoritmo a scansione lineare di tipo tradizionale e.

La fase di "rendering" 6 prevede anche un'operazione di simulazione delle ombre che l'oggetto tridimensionale proietta sulla superficie di sfondo.

Secondo una forma esemplificativa, ma non limitativa, della presente invenzione, lo sfondo dello schermo 12 è costituito dalla pagina html visualizzata nel web browser 29 implementante il componente attivo «Player 3D» 30; l'ombra proiettata oscura parzialmente il testo o le immagini sottostanti della pagina.

Tale visualizzazione viene effettuata per mezzo di un meccanismo del tipo "alpha blending".

Insieme con il "buffer frame rgb", viene generato un ulteriore buffer, vale a dire il "buffer frame UV"; con questo processo ad ogni pixel viene associato l'identificativo dell'immagine cui il pixel appartiene e i valori delle coordinate polari UV associate a quel pixel nel sistema di riferimento dell'immagine. In tal modo, quando un pixel viene selezionato dall'utente per mezzo del mouse, il "buffer frame UV"



permette di restituire informazioni circa l'immagine e le sue coordinate polari UV in corrispondenza del punto prescelto.

Se i punti selezionati cadono all'interno dell'area rettangolare di un link attivo, come mostrato in figura 9 (riferimento 14), il sistema esegue la funzionalità del link e passa ad una nuova pagina html ripetendo le fasi del procedimento dal blocco 10 in poi.

Nella fase 13 vengono intercettati gli eventi del mouse generati dall'utente. Per ogni click del tasto sinistro viene ricostruita la posizione che avrebbe avuto il puntatore sul browser bidimensionale e quindi, se il click è avvenuto su un link attivo (riferimento 14), viene inviato un evento di navigazione al relativo browser locale in memoria o al browser principale contenente l'oggetto tridimensionale.

Se invece l'evento corrisponde ad una fase in cui il mouse viene spostato con il tasto sinistro premuto (riferimento 15), il «Player 3D» 30 fa ruotare l'oggetto coerentemente con i gradi di libertà impostati dall'autore o, nel caso l'utente abbia premuto il tasto Shift o Ctrl, l'oggetto viene rispettivamente traslato o scalato (riferimento 15 di figura 9). Qualsiasi degli eventi succitati comporta la necessi-

tà di ridisegnare l'oggetto tridimensionale sullo schermo 12; a ciascun evento descritto corrisponde infatti un cambiamento di posizione relativa dell'oggetto tridimensionale rispetto al browser di sfondo che lo contiene.

L'esigenza di effettuare una nuova visualizzazione dell'immagine a video è rappresentata nella figura 9 dal riferimento 15 che riporta il controllo del flusso alla fase di "rendering" 6.

Se l'utente preme, invece, il tasto destro del mouse, il sistema visualizza un menu contestuale che propone, per esempio, di far partire e/o fermare l'animazione eventualmente associata, di aprire un nuovo browser con lo stesso contenuto della faccia sottostante, di avanzare o indietreggiare di uno stato preferenziale (si veda la figura 11), di riportare l'oggetto al suo stato iniziale, di attivare un filtro per migliorare la leggibilità del contenuto.

In genere i contenuti informativi di testo soffrono di problemi di leggibilità se non sono visualizzati in piano come su carta stampata.

Nel caso in questione il contributo testuale può apparire in svariate situazioni di orientamento: su una faccia laterale dell'oggetto o inclinato rispetto al piano del monitor 12.

In più è soggetto alla distorsione prospettica insita nella simulazione della visione di oggetti tridimensionali. Per consentire anche in queste condizioni estreme una buona visualizzazione del messaggio il componente «Player 3D» 30 implementa un filtro di contrasto che ha l'effetto di evidenziare i particolari della pagina con una ricaduta positiva sulla leggibilità dei testi.

Il meccanismo di funzionamento della fase di "rendering" 6 merita una attenzione particolare e una trattazione dedicata.

Poiché scopo principale della presente invenzione è introdurre un nuovo concetto di interfaccia tridimensionale per la navigazione in Internet, è fondamentale che la soluzione proposta garantisca un elevato grado di interattività con l'utente 13.

Il collo di bottiglia in tutte le applicazioni tridimensionali in tempo reale è la fase di "rendering" 6. Il sistema oggetto della presente invenzione ha prestato particolare attenzione all'ottimizzazione delle prestazioni del un motore di "rendering". I dettagli della procedura di "rendering" 6 sono ampiamente esposti in figura 10.

Il flusso di informazioni 16 all'ingresso del procedimento di "rendering" 16A è costituito dai dati geo-

metrici dell'oggetto tridimensionale (numero e dimensione dei poligoni che ne definiscono la superficie esterna); daidati cinematici relativi alla posizione dell'oggetto (valori di traslazione, rotazione e scala) nel sistema di riferimento del componente attivo 30; dalle immagini estratte dai web browser che devono essere visualizzate sulle superfici dell'oggetto, o in alternativa dalle informazioni sul colore col quale raffigurare l'oggetto.

Il primo stadio (riferimento 17) della fase di "rendering" esegue un algoritmo di scansione lineare; in tale fase le informazioni geometriche e di posizione vengono processate e relazionate dal punto di vista dell'osservatore per ottenere l'effetto prospettico dell'oggetto tridimensionale.

Il risultato dell'operazione è costituito dalla posizione dei vertici di ciascuna faccia che compone l'oggetto, proiettata sul piano dell'immagine.

Di fatto la posizione dei vertici nello spazio tridimensionale viene trasformata nelle coordinate bidimensionali del piano immagine in modo da consentire alla procedura successiva 18 di generare un'immagine ("bitmap") della rappresentazione prospettica dell'oggetto.

Come si è fatto cenno in precedenza, la visione pro-



spettica dell'oggetto è determinata dalla posizione nello spazio dal punto di vista dell'osservatore.

Secondo la presente invenzione, l'osservatore virtuale (o camera) è posto al centro della finestra che contiene l'oggetto tridimensionale, esternamente ad essa, e con lo sguardo rivolto verso il suo centro. Tale posizione è di fatto la posizione dell'utente che osserva il monitor del computer.

Il sistema descritto è potenzialmente in grado di consentire l'animazione dell'osservatore, cioè, a condizione di fornire l'adeguata interfaccia all'utente che osserva l'oggetto, è possibile trasportare dinamicamente la camera in qualsiasi punto dello spazio tridimensionale virtuale.

Se immaginiamo oggetti tridimensionali complessi con concavità, buchi, e interconnessioni con altri oggetti e una camera da presa animata, il sistema è estendibile in maniera naturale alla navigazione di ambienti tridimensionali complessi e immersivi.

L'utente si muove cioè all'interno di spazi chiusi in cui le pagine html appaiono proiettate sulle superfici che costeggia durante la navigazione ("walk-through").

La successiva procedura 21 del motore di "rendering" 16A esegue i calcoli di illuminazione necessari per

la rappresentazione dell'oggetto; una delle condizioni fondamentali per ottenere una simulazione realistica della tridimensionalità dell'oggetto, quando questo è rappresentato sul supporto bidimensionale del piano del monitor, è che la sua superficie presenti aree illuminate in maniera non omogenea a seconda dell'esposizione delle stesse alle sorgenti di luce virtuali presenti nella scena.

Il «Player 3D» 30 simula la presenza di una fonte luminosa posta di fronte alla camera e diretta verso la scena e l'algoritmo di illuminazione si occupa di calcolare l'ombreggiatura della superficie in rapporto a tale sorgente.

Tale calcolo considera la direzione di incidenza del raggio luminoso rispetto al piano su cui giace il punto della superficie da sottoporre a "rendering" e in base a tale angolo ne modula l'intensità.

Per ovvi motivi di performance solo i vertici di ciascuna superficie sono calcolati in tal modo, mentre il valore di colore per tutti i pixel della "bitmap" è calcolato per interpolazione a partire dalle informazioni di colore calcolate ai vertici.

Il blocco 18 permette di elaborare definitivamente la scena tridimensionale, dopo che tutte le trasformazioni geometriche e i calcoli di illuminazione sono

stati eseguiti.

Tale blocco 18 esegue le procedure di "rasterizzazione", "texturing" e "blending", scritte in linguaggio macchina ("Assembly") ed ottimizzate sia per MMX (tecnologia proprietaria per processore "Pentium") che per 3DNow (tecnologia proprietaria per processore AMD).

La procedura permette di rilevare automaticamente il tipo di processore (CPU) installato sulla macchina che esegue il «Player 3D» 30, e abilita l'appropriata ottimizzazione.

Nel caso di processori sprovvisti di tecnologia di parallelizzazione hardware, delle normali routine scritte in linguaggio standard C emulano in maniera puramente software le stesse funzionalità altrimenti implementate in linguaggio macchina.

Il processo di "rasterizzazione" implementato dal blocco 18 crea una "bitmap" 19 in memoria di dimensioni adeguate, in pixel, per contenere l'immagine dell'oggetto e con la struttura dati necessaria per immagazzinare informazioni sulla profondità di colore di ciascun pixel (bitmap RGBA).

Successivamente, nella bitmap 19 vengono disegnati i poligoni che compongono la superficie dell'oggetto sulla base delle posizioni dei vertici in uscita dal

blocco 18; a tale scopo solo le coordinate bidimensionali dei vertici proiettati sul piano immagine vengono utilizzate. A questo livello, l'informazione di profondità (coordinata Z) di ciascun vertice e' pero' ancora presente per permettere di calcolare correttamente la mappatura delle immagini.

Disegnare un poligono nella bitmap 19 significa assegnare un colore a ciascun pixel che compone l'area di immagine corrispondente al poligono; se al poligono non è associata alcuna struttura, viene effettuata una procedura di "rendering" sul solo colore di sfondo ed in questo caso si ha un effetto di ombreggiatura interpolando i colori dei vertici.

Al contrario, se al poligono è associata un'immagine, i pixel del poligono assumono il colore definito dall'immagine a valle del processo di correzione prospettica. Quest'ultimo sfrutta i valori di profondità (coordinata Z) dei vertici per determinare la distorsione subita dall'immagine una volta proiettata sul piano del poligono.

Inoltre, poiché nel blocco 21 sono stati calcolati i valori di luminosità per ciascun vertice, il colore finale di ciascun pixel è ottenuto come risultato di una operazione di "blending" (miscelazione) tra il colore dell'immagine e il valore di luminosità inter-



polato dai valori di luminosità dei vertici; in tal modo, l'immagine del poligono appare come una normale pagina html ma correttamente ombreggiata in relazione alla sua posizione nello spazio.

Alle porzioni di "bitmap" 19 non contenenti l'oggetto tridimensionale è associato un valore di trasparenza tale da non oscurare le parti di pagina sullo sfondo. Lo strato trasparente è a tutti gli effetti una ulteriore immagine a 8 bit che viene composta sul canale "alpha" della "bitmap" 19 a 32 bit; l'ombra dell'oggetto è calcolata in questa fase generando un'immagine nera dell'oggetto, lievemente spostata rispetto alla sua vera posizione. Tale immagine nera viene rappresentata nella "bitmap" 19 a 32 bit con il valore di canale "alpha" atto ad ottenere la corretta composizione dell'ombra sull'immagine di sfondo.

La conclusiva procedura di miscelazione della "bitmap" a 32 bit (riferimento 19 di figura 10), precedentemente calcolata, con la "bitmap" di destinazione corrispondente alla pagina html di sfondo, è eseguita da funzioni specializzate. In particolare, per eseguire l'"alpha blending" tra due immagini, esse devono essere descritte nello stesso formato e, a seconda della scheda grafica di cui il computer dell'utente è dotato, la "bitmap" di sfondo può essere rappresenta-

ta con profondità di colore a 16, 24 o 32 bit.

Per questo motivo, per ciascun tipo di formato esiste una funzione specifica che, in prima istanza, converte la "bitmap" in memoria nel formato della "bitmap" ospitante, e che successivamente esegue l'effettiva miscelazione.

Tutte le procedure di conversione di formato sono implementate in linguaggio macchina in modo da sfruttare al meglio le caratteristiche hardware del processore; se al tipo di processore non corrisponde una procedura specializzata, interviene un'analogha procedura implementata in linguaggio di alto livello.

Oltre alla "bitmap" 19 contenente le informazioni di colore e di trasparenza (canale "alpha RGBA") di ogni pixel, il processo di "rendering" genera una seconda uscita 20, che garantisce le funzionalità di ipertesto delle pagine visualizzate sull'oggetto.

Essa è una ulteriore "bitmap", cui ci si riferisce anche col nome di "UV frame buffer", che contiene, per ogni pixel, l'identificativo dell'immagine cui appartiene e le coordinate UV del punto sull'immagine stessa. Tale informazione è utilizzata nel blocco 13 di figura 9, che gestisce le interazioni con l'utente. Quando l'utente, muovendosi con il puntatore del mouse, seleziona un pixel della struttura,

l'"UV frame buffer" fornisce il corrispondente identificativo (ID) dell'immagine e i valori delle coordinate polari UV del punto selezionato.

Il «Player 3D» 30 verifica che tali valori di coordinate UV della specifica immagine appartengano ad una delle regioni rettangolari contenenti un link attivo della pagina html associata a quell'immagine e, in tal caso, carica l'indirizzo URL corrispondente secondo lo schema di flusso 14 di figura 9. Come visto in precedenza, la lista delle posizioni delle aree rettangolari sensibili per ciascuna immagine è conservata in memoria dal blocco 11 di figura 9.

Per motivi di performance, la "bitmap" 20 viene generata dal processo di "rendering" solo ad oggetto fermo, mentre quando l'utente sta manipolando l'oggetto o semplicemente osservando la sua animazione preimpostata, tale informazione aggiuntiva non è prodotta, per il fatto che non ci si trova nell'eventualità che l'utente acceda ad un link mentre questo si muove.

Nelle figure 2a e 2b è riportata la descrizione dettagliata del formato di file da utilizzare per definire un elemento tridimensionale interpretabile dal «Player 3D» 30. In particolare, è riportata la descrizione del formato testo e di tutti i "tag" necessari alla definizione di ogni proprietà dell'oggetto.

Come si è ripetutamente detto, la comprensione di tale formato è indispensabile al programmatore del sito se intende generare componenti tridimensionali in maniera dinamica appoggiandosi a "server script"; inoltre, il formato di file binario prodotto dal sistema autore, a seguito di una composizione visuale dell'elemento e delle sue proprietà, di fatto contiene le stesse informazioni criptate descritte in chiaro nel file in formato testo.

Come risulta dalle figure 2a e 2b, la struttura del documento è organizzata a "tag", in modo tale che ciascun tag descriva una proprietà e il campo che ne segue definisca il valore di tale proprietà.

Nel caso in cui al "tag" siano associati dei dati, è necessario definirne il valore ed il loro significato. Se trattasi di valori multipli, essi devono essere separati da una virgola.

Il primo "tag" è la parola chiave ELEMENT; tale parola chiave indica il punto di partenza della descrizione di un elemento tridimensionale. Tutte le successive istruzioni definiscono proprietà di quello specifico elemento. Una ulteriore dichiarazione ELEMENT sancisce la fine della descrizione di un oggetto e l'inizio di una successiva descrizione.

La presenza della dichiarazione ELEMENT è obbligato-



ria e deve essere posta all'inizio del set di istruzioni relative alla descrizione dell'oggetto, come appare evidente nella successiva figura 3, che raccoglie alcune regole fondamentali da rispettare per generare un file nel formato corretto.

Il successivo "tag" da definire è l'identificativo dell'elemento in questione. Se la pagina html contiene istanze multiple di elementi tridimensionali è fondamentale che il «Player 3D» 30 possa indirizzare ciascuno di essi. Un'altra dichiarazione è costituita dal "tag" SIMPLIFIED, che indica la tipologia dell'oggetto tridimensionale.

Per ragioni di semplicità, nella presente descrizione funzionale, il «Player 3D» 30 è in grado di interpretare oggetti semplici, per la precisione cilindri regolari; nel linguaggio della geometria, per cilindri regolari si intendono oggetti solidi aventi per direttrice un poligono regolare e generatrici parallele alla normale al poligono. Il sistema è comunque in grado di gestire volumi complessi definiti dal "tag" COMPLEX.

Come appare dalla figura 2a, per la descrizione di un elemento di tipo SIMPLIFIED, è sufficiente indicare la dimensione del lato del poligono di base (espressa in pixel), l'altezza del poliedro (espressa in

pixel), e il numero di facce laterali dell'oggetto (o, in altre parole, il numero di lati del poligono regolare di base).

Segue una serie di "tag" di tipo PAGEID_AND_ADDRESS in cui per ogni poligono che compone la superficie dell'oggetto è possibile associare l'indirizzo di un URL da visualizzare sul poligono stesso.

La struttura della descrizione è assolutamente intuitiva ed è costituita dall'elemento identificativo del poligono e dalla stringa di testo dell'indirizzo URL da visualizzare: la convenzione adottata per gli identificativi dei poligoni risulta elaborata in modo tale che gli identificativi 0 e 1 corrispondono rispettivamente al poligono superiore ed inferiore del cilindro. Tutti gli identificativi successivi sono assegnati in sequenza ai poligoni della superficie laterale dell'oggetto.

Come risulta dalla figura 4 qualsiasi indirizzo URL valido per un web browser standard è un indirizzo valido visualizzabile sui poligoni. Nell'esempio di figura 4 compaiono indistintamente indirizzi di pagine html e di "server script asp".

I "tag" XTRANDOF, YTRANDOF e ZTRANDOF specificano i gradi di libertà offerti all'interazione con l'utente; nel caso in questione di traslazione,

l'autore decide se concedere all'utente la facoltà di traslare l'oggetto nello spazio della pagina.

Omettendo per esempio il "tag" XTRANDOF all'utente viene impedita qualsiasi interazione con l'oggetto tridimensionale nel senso della sua traslazione nella direzione dell'asse X.

In alternativa, l'autore decide se dare libertà di movimento senza limitazioni, assegnando valore 0 all'intero successivo, se limitarne la traslazione solo verso sinistra (valore 1), solo verso destra (valore 2), o in entrambe le direzioni (valore 3). Se qualche limitazione è presente, essa deve essere esplicitata in uno o due valori numerici che definiscono l'ampiezza di movimento consentito (i due valori "float" successivi).

La stessa logica descritta in dettaglio per la traslazione in X si applica per tutti gli altri gradi di libertà di traslazione, di rotazione e di scala.

Segue una sezione relativa all'animazione degli oggetti tridimensionali.

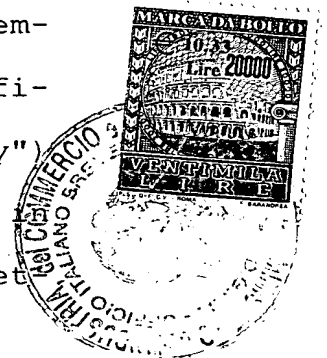
Ad ogni parametro di animazione dell'oggetto tridimensionale è possibile associare una funzione che ne descrive la dinamica nel tempo. Il modo di definire tale funzione consiste nel far precedere la sua dichiarazione dal "tag" principale FCURVETYPE (vedi

anche la regola di figura 4), che definisce il tipo di interpolazione che la funzione di animazione usa tra un "key frame" e l'altro.

I "key frame" specificati devono cioè essere interpolati da una funzione continua perché possa sempre essere restituito un valore della proprietà per ogni istante di tempo. L'interpolazione tra i "key frame" può essere di tipo lineare, "spline" o "bezier".

I "key frame" costituiscono un campionamento nel tempo dell'animazione della proprietà. L'autore specifica cioè per alcuni istanti di tempo precisi ("key") il valore che deve assumere la proprietà quell'istante. Tutti gli altri valori sono, come detto, ricavati per interpolazioni da questi ultimi.

La descrizione dei "key frame" consiste nella definizione di valori intercalati dal "tag" DOFKEY, dove DOF specifica la proprietà che si vuole animare (per esempio XTRANKEY). I valori che caratterizzano un "key frame" sono l'istante di tempo (espresso con un numero intero corrispondente alla posizione del fotogramma all'interno della sequenza animata) e il valore della proprietà in quell'istante (per esempio la posizione in X). Nel caso di interpolazione di tipo "bezier", due ulteriori valori numerici specificano la pendenza della funzione a destra e a sinistra del



"key frame" in questione.

La sezione successiva, mostrata nella figura 2b, consente di impostare stati di posizione preferenziali dell'oggetto. Quando cioè l'utente interagisce con l'oggetto tramite il mouse, esso si posiziona automaticamente in uno stato preferenziale non appena l'utente rilascia il pulsante del mouse. Lo stato preferenziale cui l'oggetto converge è naturalmente quello più vicino alla posizione dell'oggetto al momento del rilascio. In genere tali stati preferenziali sono definiti dall'autore per garantire che l'oggetto si stabilizzi in una posizione e orientamento atti ad assicurare sempre una buona visione o leggibilità delle pagine visualizzate su di esso.

La descrizione degli stati preferenziali è preceduta dal "key tag" PREFSTATE. Dopo di esso può comparire un "tag" di PREF per ogni tipo di movimento (PREF-TRAN, PREFROT, PREFSCALE). Lo stato preferenziale è costituito da tre valori numerici in sequenza che fissano i valori per la specifica proprietà (valori di X, Y e Z).

Il successivo "tag" SCENEBGCOLOR definisce il colore di sfondo.

Come precedentemente accennato, l'oggetto tridimensionale si muove all'interno dell'area rettangolare

del componente attivo che lo contiene.

In genere tale area è trasparente e quindi non visibile. Per particolari esigenze è tuttavia possibile specificare un colore che farà da sfondo all'oggetto tridimensionalee in questo caso la regione del componente attivo si sovrappone alla pagina html sottostante.

Questa proprietà acquista particolare senso quando si pubblicano gli oggetti tridimensionali in modalità "Window".

In modalita' "Window" l'oggetto tridimensionale è visualizzato e si muove all'interno di uno specifico spazio della pagina web, definito dall'utente. Tale spazio che è parte integrante della pagina, è atto a contenere e limitare il movimento dell'oggetto e possiede un proprio colore di sfondo. In alternativa, in modalità "Windowless", l'oggetto è libero di muoversi al di sopra della la pagina sottostante e appare come un componente tridimensionale sovrapposto ad essa. La modalità di visualizzazione "Window" o "Windowless" viene specificata nel sistema autore e tradotta in istruzione HTML come risulta nella figura 6. Tale specifica e' considerata al momento del esecuzione del «Player 3D» 30.

Il "tag" FACEBGCOLOR definisce invece un colore co-

stante per quelle facce dell'oggetto sulle quali non si vuole visualizzare alcun contenuto web. In questo caso oltre al valore numerico del colore è necessario indicare l'elemento identificativo del poligono da colorare; il criterio adottato per l'identificazione dei poligoni, è lo stesso descritto in precedenza.

I valori numerici successivi alla dichiarazione del "tag" DEFAULT_TRANSFO specificano i valori iniziali di SRT ("Scaling", "Rotation", "Translation").

Al momento dello scaricamento del file all'interno di un web browser, l'oggetto tridimensionale ivi descritto apparirà nella posizione orientamento e dimensione definite da quelle coordinate.

Il "tag" STARTANIMATION consente di forzare l'esecuzione dell'animazione dell'oggetto (se esistente) non appena questi viene caricato in memoria.

Il "tag" NOAUTODEEP consente di inibire l'operazione automatica di spostamento dell'oggetto verso l'asse delle coordinate Z negative.

Lo slittamento dell'oggetto in profondità mira a far coincidere la sua faccia in primo piano con l'origine del sistema di riferimento. L'accorgimento descritto previene l'espansione della dimensione della faccia (rispetto ai valori in pixel impostati dall'autore) per l'effetto prospettico dovuto all'avvicinamento

della faccia in primo piano verso il punto di vista dell'osservatore virtuale.

Infine il "tag" TURNING consente di adagiare in orizzontale il cilindro definito sopra per ottenerne una sorta di effetto rullo. In questo caso la base e l'altezza di ciascun poligono vengono invertite in modo che la visualizzazione della pagina html subisca un'analogia rotazione.

La figura 4 è un esempio autoesplicativo di come descrivere un semplice oggetto tridimensionale con pagine web visualizzate sulla sua superficie. Esso si conforma rigidamente alle regole di formato descritte nella sezione precedente.

Le figure 7a, 7b, 8a e 8b mostrano i due differenti schemi di flusso che governano i due diversi approcci (statico e dinamico) per la generazione dei componenti tridimensionali.

I processi e i sistemi hardware e software coinvolti nei due casi sono sostanzialmente diversi e implicano da parte dell'autore attività di complessità crescente; in realtà, i due approcci comportano differenti attività solo da parte dell'autore, cioè in fase di generazione dei componenti tridimensionali, mentre, dal punto di vista dell'utente, il «Player 3D» opera allo stesso modo e solo in fase di acquisizione dei



dati, e in maniera trasparente all'utente, adotta due procedure differenti a causa della diversa natura dei dati in ingresso.

Nell'approccio statico descritto nelle figure 7a e 7b, l'elemento fondamentale è il sistema autore 31. Esso è un'applicazione software che consente al creatore delle pagine web di costruire gli oggetti tridimensionali in maniera visuale e di comporli sulla pagina html; inoltre, comuni strumenti ereditati da applicazioni tipicamente orientate al tridimensionale, consentono la definizione accurata dell'animazione dell'oggetto. Il software autore è più ampiamente illustrato nella successiva figura 11. Una volta creato l'oggetto e definite le sue proprietà visuali e dinamiche, il sistema autore produce in uscita un file contenente tutte le informazioni necessarie per ricostruire l'oggetto; il file generato risulta in formato binario.

Oltre al file binario, il sistema autore produce le istruzioni html di figura 6, le quali invocano il «Player 3D», definiscono le dimensioni del componente attivo da creare all'interno della pagina web e specificano il nome del file contenente la descrizione dell'oggetto da visualizzare (il file precedente). È cura dell'autore inserire tali istruzioni html

all'interno del codice html della pagina destinata ad ospitare il componente tridimensionale.

Il file descrittore dell'oggetto, alla pari delle pagine html che compongono il sito web, è pubblicato sul web server di destinazione; tale operazione è illustrata dal blocco 32. Le dimensioni tipiche del file binario sono dell'ordine di 1 Kb, mentre il tempo di scaricamento con un modem standard a 28.8 kbps è inferiore al secondo.

Dalla parte dell'utente, con la procedura indicata con 33 in figura 7b, l'utente accede alla pagina web contenente il componente tridimensionale.

Al momento dell'esecuzione delle istruzioni di figura 6, se il «Player 3D» 30 non è installato nel web browser prescelto, quest'ultimo si connette automaticamente ad uno specifico web server, scarica il file del «Player 3D» 30, lo decompime e lo installa.

Queste ultime due operazioni sono rappresentate nella figura 7b dal blocco di scelta 34 e dal processo 35.

Una volta installato, o se già residente nel browser, il «Player 3D» 30 viene eseguito; esso interpreta il file che descrive il componente tridimensionale e lo visualizza sul monitor. L'esecuzione del «Player 3D» 30 è rappresentata dal blocco di processo 36.

Le figure 8a e 8b illustrano lo schema di flusso

dell'approccio dinamico. Dalla parte del web server, la prima operazione a carico dell'autore è richiedere una licenza per il proprio dominio.

Diversamente dal caso precedente, la generazione del file che descrive il componente tridimensionale non si appoggia al sistema autore e, trattandosi di un comune file di testo, esso è compilabile con un comune "editor" di testo, a patto di rispettarne il formato documentato nelle figure 2a, 2b, 3 e 4.

Il «Player 3D» 30 è in grado di interpretare anche tale formato, ma, in questo caso, ne certifica la licenza. Come per il caso del file binario, le dimensioni tipiche del file di testo sono dell'ordine di 1 Kb e il tempo di scaricamento con un modem standard a 28.8 kbps è inferiore al secondo.

Per meglio illustrare l'aspetto della licenza, la figura 1 riporta il meccanismo dettagliato dell'autenticazione, che coinvolge precise procedure sia dal lato server che dal lato utente.

Nell'approccio statico, il flusso procede linearmente dal server http 45 al computer utente 48, passando attraverso la fase di scaricamento 47 del file binario 46 fino alla sua finale visualizzazione sul monitor dell'utente, tramite il componente attivo 49.

Nel caso dinamico il flusso è più articolato e coin-

volge alcuni processi aggiuntivi.

Nel server http 45 risiedono uno o più file script 51 interpretabili dal server stesso 45 e il file 52 che contiene la chiave di licenza associata allo specifico dominio Internet. Quando il computer utente 48 accede al server http 45, i file script 51 non vengono direttamente scaricati, ma eseguiti dal server stesso; ciò che viene scaricato è il risultato 53 di tale esecuzione. In genere, i file script 51 sono file di tipo .asp, .jsp, ecc. che si appoggiano a "database" server 50 e da cui estraggono informazioni relative per esempio all'utente che ha eseguito l'accesso.

Il risultato dell'esecuzione del file script è un file di testo, che descrive nel formato corretto gli elementi tridimensionali da visualizzare tramite il componente attivo 54. Il file di testo 53 così generato giunge infine, attraverso la fase di scaricamento 55, al computer utente 48, presso il quale il componente attivo 54 riconosce il formato del file in arrivo.

Nel caso in questione, prima di visualizzarne il contenuto, il componente attivo scarica il file di licenza 52 e verifica che la chiave in esso contenuta corrisponda al nome del dominio Internet secondo uno schema di criptazione prestabilito. Infine, una volta



autenticata la licenza, il processo 54 visualizza il componente tridimensionale. Le dimensioni tipiche del file di licenza sono dell'ordine dei 100 byte e il tempo di scaricamento con un modem standard a 28.8 kbps è inferiore a 0.5 secondi.

Ritornando alle figure 8a e 8b, una volta ottenuta una chiave di licenza valida per il proprio dominio, l'autore si occupa di creare i contenuti dinamici appoggiandosi a linguaggi di script tipici dei server http; l'operazione rappresentata dal blocco 38 fa sì che il server possa generare componenti tridimensionali in maniera dinamica a seconda, per esempio, dell'utente registrato o di eventi innescati dall'utente stesso durante la navigazione.

Nella fase 39, file di script e file di licenza sono immagazzinati e pubblicati nel server web.

Dalla parte utente, tutto si svolge in maniera trasparente all'utente, ma la procedura di visualizzazione passa attraverso una sequenza di processi più elaborata che nel caso statico descritto in precedenza; infatti anche in questo caso nella fase 33 l'utente accede ad una pagina web contenente un componente tridimensionale ma prima che le pagine html vengano fisicamente scaricate, il server esegue gli script server e il risultato, costituito da file di

testo correttamente formattati, è trasmesso al browser utente. Ciò avviene nel blocco di processo 40. Successivamente, avviene la stessa verifica del caso statico, sulla presenza o meno del «Player 3D» 30 e il suo eventuale scaricamento ed installazione (processi 34 e 35).

Una volta implementato, il «Player 3D» 30 identifica il tipo di formato di file nel processo 41. Trattandosi di file di tipo dinamico, nel blocco successivo 42 legge il file di licenza dal server e confronta la chiave con il nome del dominio Internet del sito in questione. Se la licenza è valida ha inizio il processo di visualizzazione 36; in alternativa (blocco 44), il «Player 3D» 30 termina l'esecuzione senza visualizzare il contenuto del file.

In figura 11 e' rappresentato in dettaglio lo schema di flusso di una tipica sessione di lavoro basata sul sistema autore (processo 31 di figura 7). In essa sono messi in evidenza i contributi da e verso l'esterno e le procedure che invece sono strettamente eseguite all'interno del sistema stesso.

Come più volte accennato, scopo del sistema autore 31 è consentire una rapida prototipizzazione dei contributi tridimensionali che l'autore intende inserire nel proprio sito web. La procedura viene svolta in un

ambiente integrato completamente visuale ed interattivo che consente la definizione delle proprietà geometriche e dinamiche dell'oggetto e la definizione dei contributi web da mappare sulla superficie dell'oggetto stesso. Il risultato dell'operazione è rappresentato da file binari e codice html facilmente integrabile nell'architettura generale del sito.

Immediatamente dopo l'apertura del sistema autore 31, l'utente ha la facoltà di caricare in memoria un progetto salvato in una precedente sessione di lavoro al fine di modificarne alcune proprietà, o in alternativa di crearne uno nuovo. In quest'ultimo caso viene proposto un percorso guidato (detto "wizard") che facilita l'utente in maniera amichevole a definire le caratteristiche principali del nuovo oggetto.

Una volta che il progetto è caricato nel sistema, il primo insieme di proprietà che necessitano di definizione sono quelle relative all'area di lavoro 57 da utilizzare nella sessione corrente. Allo scopo una finestra di dialogo consente di specificare il percorso di una cartella di lavoro su disco fisso. In essa verranno scaricati i file di "output" generati e da essa verranno prelevati file html dichiarati nella successiva fase di mapping 62. Sempre in tale finestra di dialogo l'utente può indicare se il sistema

debba interfacciarsi con un "http server" locale, e in questo caso ne va specificato il nome, affinché esegua eventuali "server-script" associati alle facce in alternativa a file html. La possibilità di eseguire dinamicamente gli script durante la sessione di lavoro è determinante per consentire il test e la validazione delle funzionalità dell'oggetto in corso di definizione. L'ultima proprietà relativa all'area di lavoro e la specificazione del nome da assegnare all'oggetto nonché al file che lo descrive.

Il successivo blocco 58 di proprietà da definire riguardano il componente attivo. Come si è detto, il componente attivo costituisce il contenitore in cui si muove l'elemento tridimensionale. Tale contenitore occupa una porzione della pagina visualizzata nel browser e generalmente si sovrappone ad essa in maniera trasparente. Una finestra di dialogo consente di definire le dimensioni del contenitore (cioè in ultima analisi le dimensioni della regione in cui è possibile animare l'oggetto) e la modalità di visualizzazione. Due diverse modalità sono supportate: quella basata su finestra ("window-based") e quella senza finestra ("windowless"). In modalità "window-based", il riquadro in cui l'oggetto è visualizzato è ben distinguibile all'interno della pagina html e ad



esso è dedicato una specifica porzione all'interno dell'architettura della pagina stessa. In modalità "windowless", il riquadro resta invisibile (in quanto trasparente) e l'oggetto tridimensionale appare galleggiare sopra i contributi di testo della pagina sottostante. Nel primo caso la finestra di dialogo consente di definire i colori di sfondo del riquadro in questione.

Nel modulo 59 si definiscono le proprietà geometriche dell'oggetto. Nel prototipo in questione i solidi supportati sono semplici cilindri geometrici regolari per i quali restano da definire: numero di facce (cioè numero di lati del poligono regolare di base), lunghezza del lato del poligono di base e altezza del poliedro. Nella stessa finestra di dialogo è possibile specificare anche il tipo di sviluppo (orizzontale o verticale) del cilindro stesso.

L'interfaccia grafica di cui il sistema autore 31 è dotato, per la definizione delle proprietà dinamiche dell'oggetto è più elaborata delle precedenti. Essa si compone di un primo blocco 61 in cui per ciascun grado di libertà (traslazione, rotazione e scala), è possibile digitare il valore numerico della coordinata. In alternativa, posizione e orientamento dell'oggetto sono pilotabili in maniera interattiva

nella finestra di previsualizzazione, dove è possibile agganciare l'oggetto con il puntatore del mouse e trasportarlo, ruotarlo, avvicinarlo o allontanarlo nella configurazione desiderata. La finestra di dialogo contenente i valori correnti dei gradi di libertà fornisce una sezione in cui i suddetti valori possono essere impostati come valori di partenza (valori di default) al momento del trasferimento dell'oggetto dalla rete al "browser web". Ad ogni valore numerico che rappresenta la coordinata di una proprietà cinematica dell'oggetto è associato un bottone per l'inserimento dei "key frame". Alla pressione del tasto, nella curva di funzione della proprietà in questione, viene inserito un "key frame" (cioè un campione) nell'istante in cui si trova il puntatore sulla barra dei tempi e con il valore corrente assunto dalla proprietà al momento della pressione del tasto. In altre parole la procedura di inserimento dei "key frame" è un'operazione iterativa che si svolge secondo i seguenti passi: posizionamento dell'oggetto nella configurazione desiderata, inserimento del "key frame", spostamento dell'indicatore del tempo in un nuovo istante, spostamento dell'oggetto in una nuova posizione, inserimento di un nuovo "key" e così via. L'intera procedura avviene nel blocco 61, in cui si

assegna anche la proprietà di "auto start", cioè se un'animazione dell'oggetto è stata preimpostata, quest'ultimo si metterà in movimento non appena caricato nel browser dell'utente.

La definizione dell'animazione dell'oggetto può essere ulteriormente raffinata nel blocco 60. Una dedicata finestra di dialogo visualizza per ogni proprietà cinematica la corrispondente curva di funzione. Essa è rappresentata in un sistema cartesiano avente in ascissa l'asse dei tempi e in ordinata il valore della proprietà. I "key frame" sono visibilmente rappresentati tramite marcatori disegnati sulla curva che rappresenta l'andamento della proprietà. Le curve suddette sono manipolabili direttamente dal loro grafico, il quale consente in maniera interattiva di rimuovere "key frame" preesistenti, inserirne di nuovi o cambiarne la curvatura e il tipo di interpolazione tra "key frame" successivi.

Definite le proprietà cinematiche dell'oggetto, il blocco successivo 62 consente di specificare gli indirizzi URL da mappare su ciascuna faccia del poliedro. Anche in questo caso una dedicata finestra di dialogo consente una facile definizione della pagina html da utilizzare o addirittura di accedere alla rete per associare pagine di altri siti presenti in In-

ternet.

Nella pratica sarà cura dell'autore scrivere il codice html della pagina che vuole mappare sulla faccia dell'oggetto e questo avviene con comuni strumenti di "web content management" residenti esternamente al sistema autore (blocco 63).

Nella finestra di previsualizzazione è possibile selezionare in sequenza le faccia da mappare con un semplice click del mouse, ed associargli l'indirizzo desiderato. Come si è detto, a ciascuna faccia è associabile un qualsiasi valido indirizzo interpretabile dal browser, il che significa anche indirizzi di componenti attivi (i "server-script"). Nella stessa finestra di dialogo dedicata al mapping è possibile definire in alternativa all'indirizzo URL, un colore di sfondo. Un'ulteriore finestra di dialogo implementa la procedura 64, in cui l'autore ha facoltà di definire quali saranno le libertà di movimento concesse all'utente nella fase finale di interazione con l'oggetto tridimensionale visualizzato sul suo browser. Per ovvi motivi di design l'autore può voler limitare alcuni movimenti allo scopo di impedire che l'oggetto si sovrapponga a zone non desiderate della pagina sottostante oppure decidere che, per una corretta visualizzazione delle informazioni che vuole



trasmettere, l'oggetto debba limitarsi per esempio alla sola rotazione o alla sola traslazione.

La finestra di dialogo suddetta consente per ciascuna proprietà cinematica di definire se sarà modificabile dall'utente e, in tal caso, di specificarne i limiti inferiore e superiore di variabilità.

Inoltre nella stessa finestra sono impostabili gli stati di riposo preferenziali. Essi rappresentano gli stati di posizione, orientamento e vicinanza cui l'oggetto tenderà non appena rilasciato dall'utente che vi interagiva. Come descritto precedentemente, l'utente per mezzo del mouse è in grado di manovrare l'oggetto a suo piacimento e quando interrompe l'azione invece di arrestarsi nella posizione corrente, l'oggetto automaticamente si porta nello stato preferenziale più prossimo alla posizione corrente. Naturalmente la scelta degli stati preferenziali è libera ma di norma coincide con quelli che consentono la più facile leggibilità dei contenuti informativi mappati sulle facce dell'oggetto.

Il blocco di "preview" 65 è una finestra grafica in cui l'autore è in grado di verificare funzionalità e validità del lavoro in corso. In realtà, tale finestra di preview è divisa in due sezioni: la prima è propriamente l'area di lavoro dell'autore, ove

l'utente manipola liberamente l'oggetto e ne definisce interattivamente l'animazione, mentre la seconda è un effettivo preview del risultato finale che apparirà all'utente del web e consente lo stesso grado di interattività del componente attivo in mano all'utente finale. L'interazione con l'oggetto è cioè soggetta ai vincoli imposti dall'autore.

Al fine di consentire un "preview" realistico del risultato finale, il sistema autore 31 genera internamente una pagina html completa contenente al suo interno l'invocazione al componente attivo 30 e la definizione del file descrittore dell'oggetto. Inoltre crea un web browser indipendente, lo visualizza, e lo inizializza con l'indirizzo della pagina suddetta.

Nella parte inferiore dell'interfaccia grafica, un insieme di comandi 66 consentono la gestione del "playback" dell'animazione allo scopo di previsualizzarne il risultato. Oltre ai comuni controlli del "play", "rewind" e "fast forward" è presente una barra dei tempi ed un puntatore scorrevole sopra di essa. Esso è lo strumento che consente di variare il riferimento temporale per l'inserimento dei "key frame" in successivi istanti di tempo. Esso risulta anche un comodo strumento interattivo per la visualizzazione della sequenza animata in quanto l'indicatore del

tempo può essere agganciato col pulsante del mouse e trascinato sulla barra lungo l'asse dei tempi.

Due importanti proprietà di questa sezione sono quelle che definiscono la durata e la fluidità della sequenza: la durata è un valore espresso in secondi e rappresenta il tempo col quale si vuole che l'oggetto esegua l'animazione assegnatagli, mentre la fluidità esprime il grado di continuità con cui l'oggetto si muove. In pratica è una misura indiretta del numero di volte che la "bitmap", rappresentante l'oggetto, deve essere ridisegnata e ricomposta sulla pagina sottostante nell'unità di tempo per ottenere l'effetto del movimento; l'evento di riscrittura della "bitmap" corrisponde ad un ciclo di "rendering" nel componente attivo 30.

Una volta completata la fase di definizione delle proprietà e validato il lavoro eseguito nella finestra di "preview", l'ultimo passo consiste nell'esportare il risultato fuori dal sistema autore 31. Ciò avviene nel blocco 67 di figura 11.

L'effetto dell'esportazione consiste nella generazione di un file binario 69 che contiene al suo interno in maniera criptata tutte le informazioni che l'autore ha specificato per l'oggetto in questione e un insieme di istruzioni html 68 (dettagliatamente

riportate nella figura 6), che l'autore provvederà ad inserire nella pagina html destinata ad ospitare l'oggetto tridimensionale. Entrambi i contenuti sono infine pubblicati nel web server destinato a contenere il sito web in costruzione.

Dalla descrizione effettuata risultano chiare le caratteristiche del sistema per la creazione, la visualizzazione e la gestione di oggetti tridimensionali su pagine web e del metodo relativo, che sono oggetto della presente invenzione, così come chiari ne risultano i vantaggi.

E' chiaro, infine, che numerose altre varianti possono essere apportate a tale sistema e a tale metodo senza per questo uscire dai principi di novità insiti nell'idea inventiva, così come è chiaro che, nella pratica attuazione dell'invenzione, le funzioni, i materiali, le forme e le dimensioni dei dettagli illustrati potranno essere qualsiasi a seconda delle esigenze e gli stessi potranno essere sostituiti con altri tecnicamente equivalenti.

Ing. Barzanò & Zanardo Milano S.p.A.



RIVENDICAZIONI

1. Metodo per la creazione, la visualizzazione e la gestione di oggetti tridimensionali su pagine web, atto a consentire di accedere ad oggetti tridimensionali presenti su un web browser (29) di una rete del tipo World-Wide-Web, caratterizzato dal fatto di comprendere una prima fase di elaborazione elettronica di detti oggetti, che vengono trattati in modo tale da apparire come oggetti reali su pagine standard del web, detti oggetti essendo atti ad interagire con un utilizzatore, in modo tale da visualizzarne informazioni presenti su loro superfici esterne, ed una seconda fase di sovrapposizione di pagine html su dette superfici degli oggetti, in modo tale che ciascuna faccia di ogni oggetto si comporti come un web browser (29) standard, supportandone le funzionalità.
2. Sistema per la creazione, la visualizzazione e la gestione di oggetti tridimensionali su pagine web implementante un metodo relativo come alla rivendicazione 1, caratterizzato dal fatto di comprendere almeno due soggetti attori ed almeno due diverse configurazioni "hardware", in cui un primo sistema comprende un web server (22), ove sono immagazzinate le pagine web contenenti detti oggetti tridimensionali, una serie di chiavi di accesso (25, 26), un computer

(22A) ed un programma applicativo per elaboratori atto a generare le pagine web, ed un secondo sistema comprende mezzi per l'esecuzione di web browser (29) su uno o più elaboratori (24) di utenti finali, almeno una applicazione software (30) funzionante nel web browser (29) ed almeno un server web (23), a cui accede l'utilizzatore per scaricare i contenuti di almeno una applicazione web (28).

3. Sistema come alla rivendicazione 2, caratterizzato dal fatto che detta applicazione software (30) è costituita dal componente attivo «Player 3D», il quale, se già presente all'interno di detto elaboratore (24) dell'utente finale, permette la generazione di un file di riferimento contenente la descrizione di detti oggetti tridimensionali da visualizzare.

4. Sistema come alla rivendicazione 2, caratterizzato dal fatto che detta applicazione software (30), costituita dal componente attivo «Player 3D», è scaricabile automaticamente dal server web (23) ed installabile nell'elaboratore (24) dell'utente finale, in modo tale che una pluralità di web browser (29) accedano a detta applicazione web (28) simultaneamente.

5. Sistema come alla rivendicazione 2, caratterizzato dal fatto che detto web browser (29) comprende almeno una interfaccia tridimensionale, che è gestibile in

modalità statica, vale a dire generata dagli autori di un sito del web, o dinamicamente composta da un sistema server, a partire da almeno una base di dati o "database" (27), per la generazione di pagine html.

6. Sistema come alla rivendicazione 2, caratterizzato dal fatto che detti oggetti tridimensionali sono creati ed animati dagli autori del sito staticamente o in maniera dinamica, eseguendo un programma server ogniqualevolta si accede ad una pagina web.

7. Sistema come alla rivendicazione 2, caratterizzato dal fatto che detto web browser (29) esegue, quale applicazione Windows, una applicazione di tipo "ActiveX", la cui finestra di visualizzazione si sovrappone ad una pagina web sottostante, detta finestra di visualizzazione comprendendo un'area, in cui non è presente detto oggetto tridimensionale da rappresentare, che è resa completamente trasparente tramite una fase di "rendering", in modo da non oscurare la pagina sottostante.

8. Sistema come alla rivendicazione 4, caratterizzato dal fatto che detto componente attivo «Player 3D» consente all'utente di manipolare la posizione e l'orientamento di detto oggetto tridimensionale nello spazio e garantisce una serie di operazioni che l'utente compie su interfacce basate su "Hypertext".

9. Metodo di creazione, visualizzazione e gestione come alla rivendicazione 1, caratterizzato dal fatto di prevedere le seguenti ulteriori fasi:

- riconoscimento (1) di formati di dati in ingresso;
- validazione (8), per mezzo di un file di autenticazione, e verifica tra chiavi di accesso, nel caso in cui si riconosca un formato testo;
- lettura (3) della descrizione di elementi da almeno un documento in rete (4, 5) e creazione di almeno un oggetto tridimensionale in memoria;
- realizzazione (9) di un nuovo web browser (29) per ogni faccia di detto oggetto tridimensionale cui è associato un indirizzo di un file html o di una applicazione server che restituisce un file html;
- caricamento delle pagine html;
- successiva implementazione di una fase di "rendering" (6) e di "rests listening" per un successivo ingresso (13) immesso dall'utilizzatore;
- implementazione di una fase di "texturing" (11), in cui viene memorizzata l'immagine di detta pagina html visualizzata nel web browser (29);
- derivazione di una immagine fisica della pagina web da detto web browser (29) a istanti consecutivi di tempo ed invio di detta immagine ad un blocco di "rendering" (6) insieme con la descrizione geometrica



e fisica di detto oggetto tridimensionale allo stesso istante, detto blocco di "rendering" (6) essendo in grado di generare e visualizzare un'immagine prospettica di detto oggetto su un monitor (12) 10. Metodo come alla rivendicazione 9, caratterizzato dal fatto che detta descrizione contiene informazioni relative alla geometria e all'animazione di detto oggetto tridimensionale del componente e gli indirizzi URL di ogni elemento html presente su ciascuna superficie di detto oggetto tridimensionale.

11. Metodo come alla rivendicazione 9, caratterizzato dal fatto che detti elementi comprendono il numero di lati, la lunghezza di ogni lato, le dimensioni delle superfici, le animazioni, le modalità preferenziali e i gradi di libertà di ogni oggetto tridimensionale.

12. Metodo come alla rivendicazione 9, caratterizzato dal fatto che detta fase di "rendering" (6) utilizza i dati geometrici di detto oggetto tridimensionale e i dati cinematici relativi alla posizione di detto oggetto in un determinato sistema di riferimento di detto componente attivo, detti dati essendo prelevati da immagini estratte da detti web browser (29) da visualizzare sulle superfici dell'oggetto.

13. Metodo come alla rivendicazione 12, caratterizzato dal fatto che detta fase di "rendering" (6) com-

prende un primo stadio (17), che esegue un algoritmo di scansione lineare, durante il quale vengono processate le informazioni geometriche e di posizione di detto oggetto tridimensionale e le stesse vengono relazionate dal punto di vista dell'osservatore per ottenere un effetto prospettico di detto oggetto.

14. Metodo come alla rivendicazione 13, caratterizzato dal fatto che detta fase di "rendering" (6) comprende un secondo stadio (18) in cui si genera un'immagine o "bitmap" (19) della rappresentazione prospettica dell'oggetto, determinata dalla posizione nello spazio dal punto di vista dell'osservatore.

15. Metodo come alla rivendicazione 14, caratterizzato dal fatto che detta fase di "rendering" genera una ulteriore uscita (20), che garantisce le funzionalità di ipertesto delle pagine visualizzate su ogni oggetto tridimensionale.

16. Metodo come alla rivendicazione 11, caratterizzato dal fatto che detta fase di animazione degli oggetti tridimensionali è effettuata in modo tale che ad ogni parametro di animazione di ciascun oggetto tridimensionale è associabile una funzione che ne descrive la dinamica nel tempo.

17. Metodo come alla rivendicazione 1, caratterizzato dal fatto che detti oggetti tridimensionali presenta-

no stati di posizione impostabili dall'utente, tramite interazione con il mouse.

18. Metodo come alla rivendicazione 1, caratterizzato dal fatto che detti componenti tridimensionali sono generati in modalità statica o dinamica.

19. Metodo come alla rivendicazione 18, caratterizzato dal fatto che detta modalità statica prevede il funzionamento di una applicazione software (31), che consente ad un autore di pagine web di costruire gli oggetti tridimensionali in maniera visuale e di comporli su una pagina html, detta applicazione producendo in uscita un file in formato binario contenente le informazioni necessarie per ricostruire detto oggetto tridimensionale.

20. Metodo come alla rivendicazione 18, caratterizzato dal fatto che detta modalità dinamica è effettuata (38) da detto server, che genera componenti tridimensionali a seconda, per esempio, dell'utente registrato o di eventi innescati dall'utente stesso durante la navigazione.

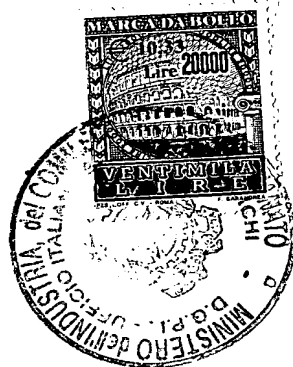
21. Metodo come alla rivendicazione 18, caratterizzato dal fatto che detta modalità dinamica si effettua tramite una fase (33) in cui l'utente accede ad una pagina web contenente un oggetto tridimensionale e, prima che dette pagine html vengano fisicamente sca-

ricate, il server esegue gli script server e il risultato, costituito da file di testo formattati, viene trasmesso (40) ad un browser utente.

22. Metodo come alla rivendicazione 21, caratterizzato dal fatto che detto componente attivo «Player 3D» legge (41) il tipo di formato di file, legge (42) un file di licenza da detto server e confronta una chiave di accesso con il nome di dominio Internet del sito e, nel caso in cui detta licenza sia valida, inizia un procedimento di visualizzazione (36), mentre, nel caso in cui detta licenza non sia valida (44), termina l'esecuzione senza visualizzare il contenuto di detto file di licenza.

23. Metodo come alle rivendicazioni 3 e 5, caratterizzato dal fatto di prevedere la realizzazione di un formato proprietario di file per la descrizione di componenti tridimensionali, rendendo detto componente attivo in grado di interpretare file, in formato binario o in formato testo, aderenti ad una prefissata formattazione, in modo tale da mettere gli autori web in grado di sviluppare applicazioni script utilizzando detta interfaccia tridimensionale.

24. Metodo come alla rivendicazione 1, caratterizzato dal fatto di prevedere la realizzazione di un sistema software (31) che consente la generazione di oggetti



tridimensionali e la definizione delle loro proprietà in un ambiente visuale ed interattivo, il quale non richiede la scrittura di codice HTML da parte dell'autore ma genera automaticamente le istruzioni di codice necessarie alla pubblicazione di detti oggetti su un sito web.

25. Sistema per la creazione, la visualizzazione e la gestione di oggetti tridimensionali su pagine web e metodo relativo come sostanzialmente descritti ed illustrati e per gli scopi specificati.

Ing. Barzanò & Zanardo Milano S.p.A.

BR/br

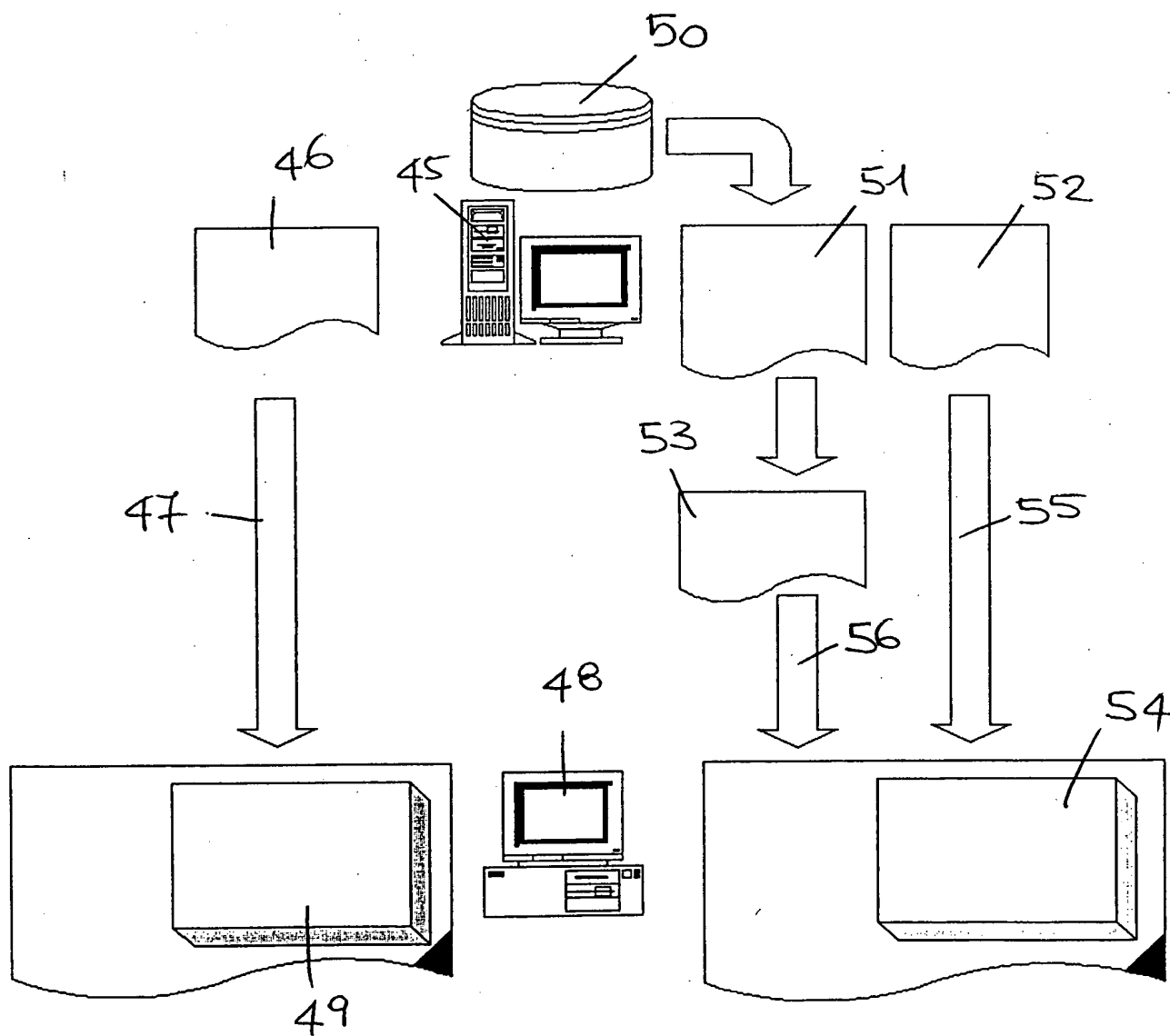
I MANDATARI

(firma)

(per sé e per gli altri)



Fig. 1



MI 2001A000538

I MANDATARI:

(firma) *W. de Felice*
(per sé e per gli altri)

Fig. 2a

```

<ELEMENT> // Dichiarazione di un elemento 3D

<ID> // Identificatore dell'elemento 3D
    integer (ID value)

<SIMPLIFIED> // L'elemento 3D e' di tipo semplificato (un poligono regolare
    estruso)
    integer (base in pixel)
    integer (altezza in pixel)
    integer (numero di facce)

<PAGEID_AND_ADDRESS> // Sul poligono identificato dall'ID sara' proiettata la pagina
    corrispondente all'URL specificato
    integer (ID del poligono)
    string (URL della pagina)

<XTRANDOF> // Gradi di liberta' per traslazione
    integer (0 traslazione libera; 1 il valore seguente sara' usato come limite inferiore; 2
    il valore seguente sara' usato come limite superiore; 3 i due valori seguenti
    saranno usati come limiti inferiore e superiore per la traslazione)
    float (minimo valore di traslazione)
    float (massimo valore di traslazione)
<YTRANDOF>
...
<ZTRANDOF>
...
<XROTDOF> // Gradi di liberta' per rotazione
    integer (0 rotazione libera; 1 il valore seguente sara' usato come limite inferiore; 2
    il valore seguente sara' usato come limite superiore; 3 i due valori seguenti
    saranno usati come limiti inferiore e superiore per la rotazione)
    float (minimo valore di rotazione)
    float (massimo valore di rotazione)
<YROTDOF>
...
<ZROTDOF>
...
<XSCALDOF> // Gradi di liberta' per scala
    integer (0 traslazione libera; 1 il valore seguente sara' usato come limite inferiore; 2
    il valore seguente sara' usato come limite superiore; 3 i due valori seguenti
    saranno usati come limiti inferiore e superiore per la scala)
    float (minimo valore di scala)
    float (massimo valore di scala)
<YSCALDOF>
...
<ZSCALDOF>
...

<FCURVETYPE> // Tipo di curva di animazione
    integer (1 lineare, 2 spline, 3 bezier)

<XTRANKEY> // Chiavi di una curva di traslazione
    integer (istante in cui la chiave e' impostata)
    float (valore di traslazione)
    float (solo nel caso di curva bezier: definisce la pendenza a sinistra della chiave)
    float (solo nel caso di curva bezier: definisce la pendenza a destra della chiave)
<YTRANKEY>
...

```

M 200 1 A 000 538

(firma) 
 (per se e per gli altri)

MANDATORY TAG:

<ELEMENT>

<ID>

altrimenti viene usato il valore 0 per ogni elemento

<SIMPLIFIED>

TAG RELATION:

<ELEMENT> prima di ogni insieme di tag legato alle proprietà dell'elemento 3D.

L' **<FCURVETYPE>** deve essere definito prima di ogni **<.....KEY>**

Fig. 3

<ELEMENT>
<ID>
1
<SIMPLIFIED>
224, 130, 8
<PAGEID_AND_ADDRESS>
0, 3d/Promo.asp
<PAGEID_AND_ADDRESS>
1, 3d/ScoopBuster.asp
<PAGEID_AND_ADDRESS>
2, 3d/MailingList.asp
<PAGEID_AND_ADDRESS>
3, <http://www.repubblica.it>
<PAGEID_AND_ADDRESS>
4, <http://www.ilsole24ore.com>
<YROTDOF>
0, 360, 0
<FCURVETYPE>
3
<YROTKEY>
0, 0, 0, 0
<YROTKEY>
100, -360, 0, 0
<START_ANIMATION>
<AUTODEEP>
<SCENEBCOLOR>
11105615

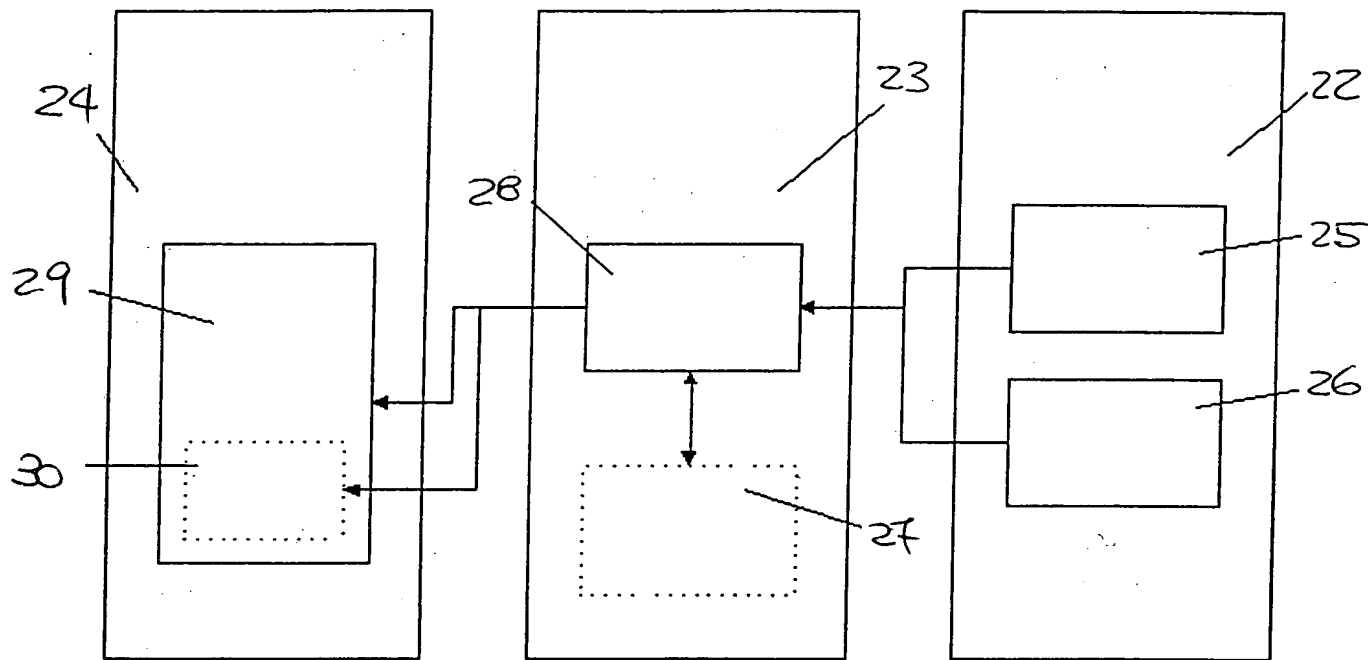
Fig. 4

I MANDATARI:

(firma)

Wendell
(per se e per gli altri)

Fig. 5



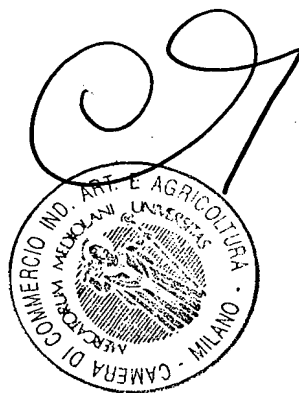
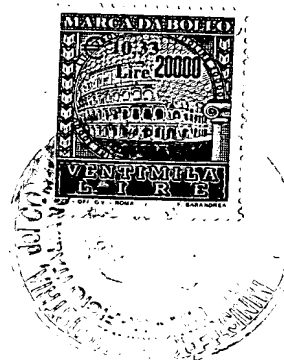
I MANDATARI

(firma)

[Handwritten signature]
(per sé e per gli atti)

```
<object classid="CLSID:11E6FE20-C5E2-11D4-B07A-0050DA2873C1"
width="120" height="120" id="element3D">
  <PARAM NAME="FileName" Value="element.wzt">
  <PARAM NAME="WLess" Value="1">
</object>
```

Fig. 6



I MANDATARI

(firma)

Waldo Lelli
(per se e per gli altri)

Fig. 7a

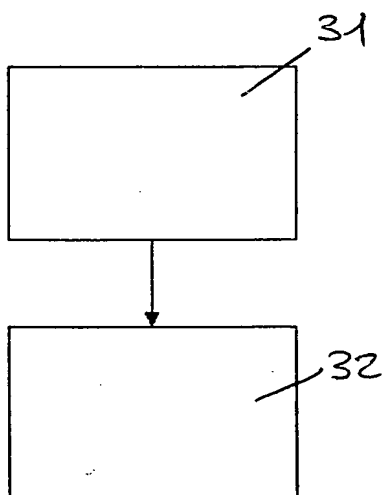
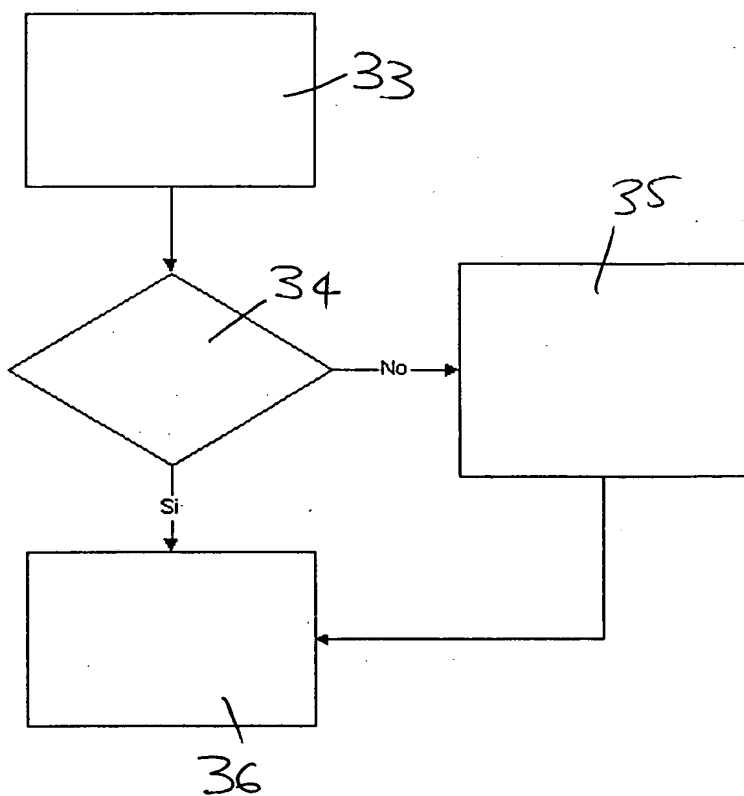


Fig. 7b



M 200 1 A 0 0 0 5 3 8



MANITARI
(firmato) *Manitari*
(per sé e per gli altri)

Fig. 8a

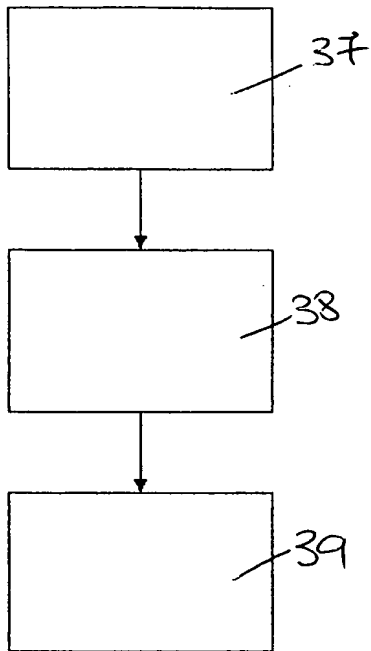
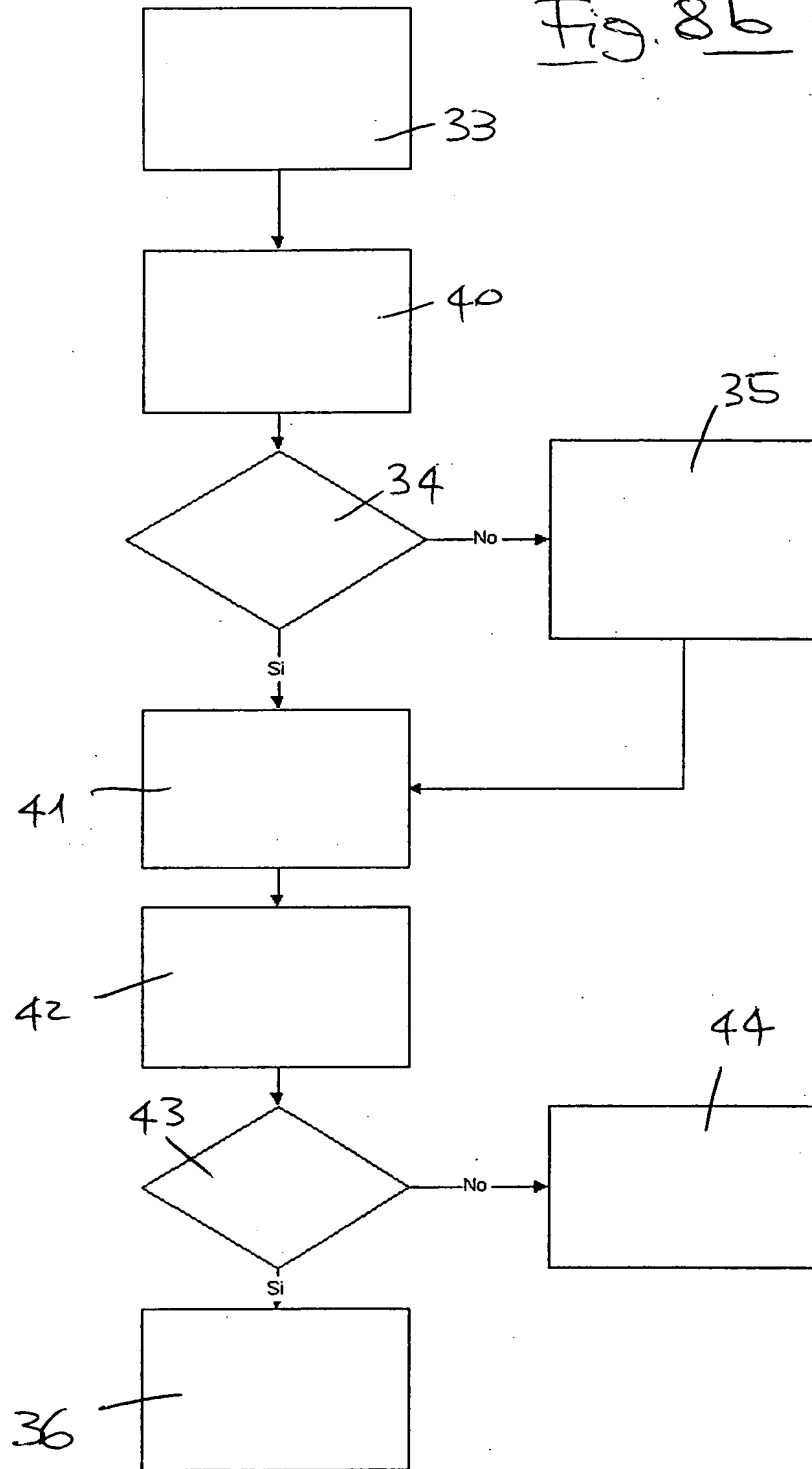


Fig. 8b

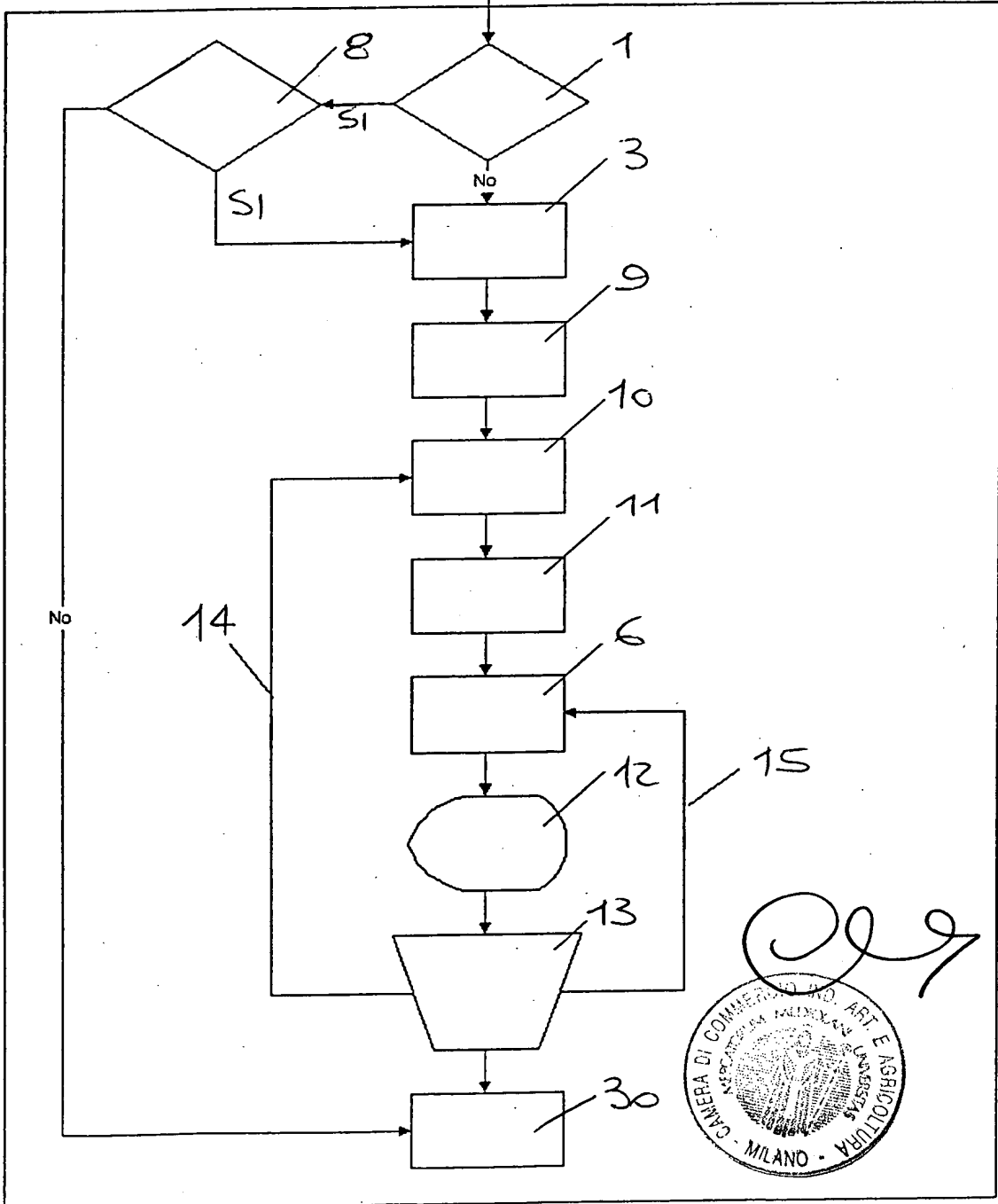
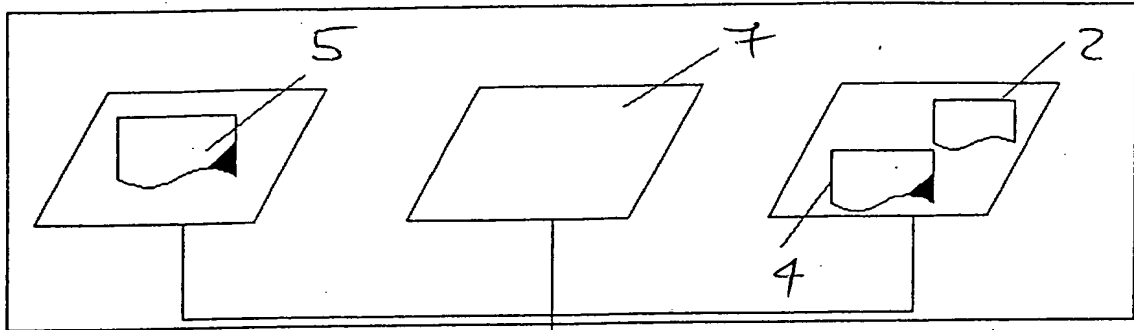


MI 200 1A 000 538



I MANDATARI:
(11/25) *Wendel*
(11/25) *Wendel*

Fig. 9

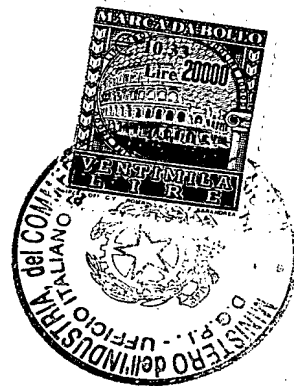
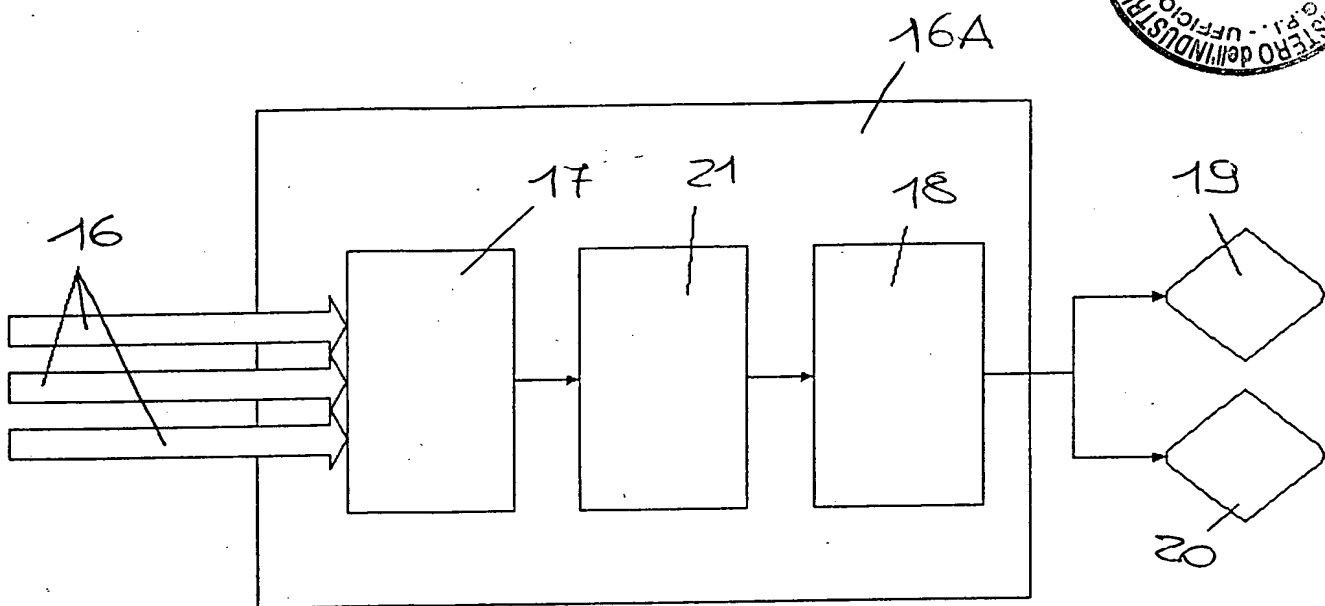


MI 200 1A 000 538

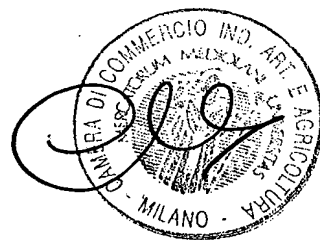
1 NOV 1974
(CHINA) *Niccolò Lelli*



Fig. 10



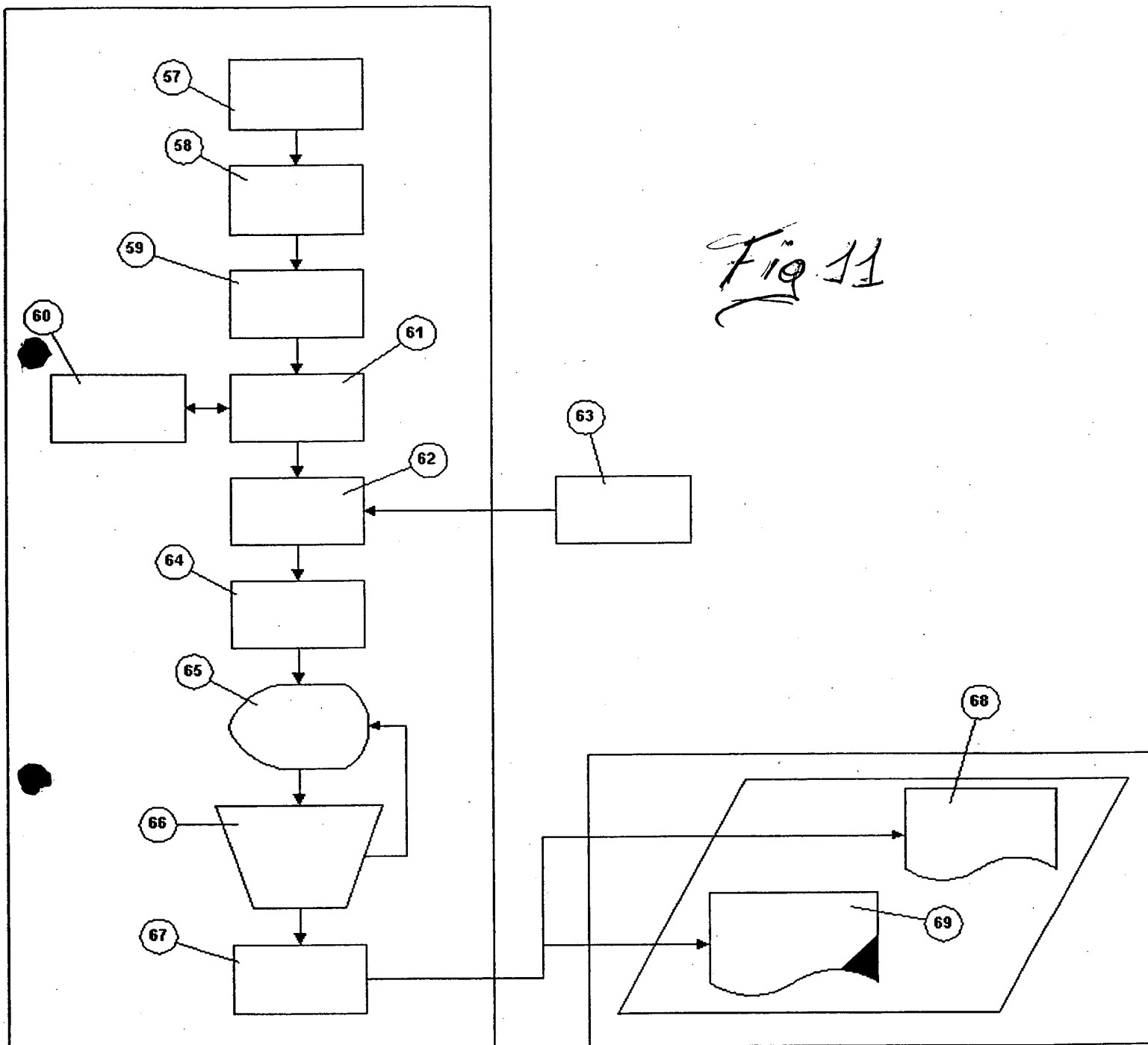
MI 200 1A 000 538



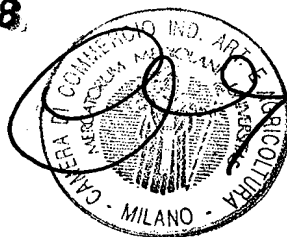
12345678910111213141516171819202122232425262728293031323334353637383940414243444546474849505152535455565758596061626364656667686970717273747576777879808182838485868788899091929394959697989910010110210310410510610710810911011111211311411511611711811912012112212312412512612712812913013113213313413513613713813914014114214314414514614714814915015115215315415515615715815916016116216316416516616716816917017117217317417517617717817918018118218318418518618718818919019119219319419519619719819920020120220320420520620720820921021121221321421521621721821922022122222322422522622722822923023123223323423523623723823924024124224324424524624724824925025125225325425525625725825926026126226326426526626726826927027127227327427527627727827928028128228328428528628728828929029129229329429529629729829930030130230330430530630730830931031131231331431531631731831932032132232332432532632732832933033133233333433533633733833934034134234334434534634734834935035135235335435535635735835936036136236336436536636736836937037137237337437537637737837938038138238338438538638738838939039139239339439539639739839940040140240340440540640740840941041141241341441541641741841942042142242342442542642742842943043143243343443543643743843944044144244344444544644744844945045145245345445545645745845946046146246346446546646746846947047147247347447547647747847948048148248348448548648748848949049149249349449549649749849950050150250350450550650750850951051151251351451551651751851952052152252352452552652752852953053153253353453553653753853954054154254354454554654754854955055155255355455555655755855956056156256356456556656756856957057157257357457557657757857958058158258358458558658758858959059159259359459559659759859960060160260360460560660760860961061161261361461561661761861962062162262362462562662762862963063163263363463563663763863964064164264364464564664764864965065165265365465565665765865966066166266366466566666766866967067167267367467567667767867968068168268368468568668768868969069169269369469569669769869970070170270370470570670770870971071171271371471571671771871972072172272372472572672772872973073173273373473573673773873974074174274374474574674774874975075175275375475575675775875976076176276376476576676776876977077177277377477577677777877978078178278378478578678778878979079179279379479579679779879980080180280380480580680780880981081181281381481581681781881982082182282382482582682782882983083183283383483583683783883984084184284384484584684784884985085185285385485585685785885986086186286386486586686786886987087187287387487587687787887988088188288388488588688788888989089189289389489589689789889990090190290390490590690790890991091191291391491591691791891992092192292392492592692792892993093193293393493593693793893994094194294394494594694794894995095195295395495595695795895996096196296396496596696796896997097197297397497597697797897998098198298398498598698798898999099199299399499599699799899910001001100210031004100510061007100810091010101110121013101410151016101710181019102010211022102310241025102610271028102910301031103210331034103510361037103810391040104110421043104410451046104710481049105010511052105310541055105610571058105910601061106210631064106510661067106810691070107110721073107410751076107710781079108010811082108310841085108610871088108910901091109210931094109510961097109810991100110111021103110411051106110711081109111011111112111311141115111611171118111911201121112211231124112511261127112811291130113111321133113411351136113711381139114011411142114311441145114611471148114911501151115211531154115511561157115811591160116111621163116411651166116711681169117011711172117311741175117611771178117911801181118211831184118511861187118811891190119111921193119411951196119711981199120012011202120312041205120612071208120912101211121212131214121512161217121812191220122112221223122412251226122712281229123012311232123312341235123612371238123912401241124212431244124512461247124812491250125112521253125412551256125712581259126012611262126312641265126612671268126912701271127212731274127512761277127812791280128112821283128412851286128712881289129012911292129312941295129612971298129913001

1922

Wendell



M 200 1 A 000 538



Wendel